Spectral tools for Dynamic Tonality and audio morphing

Spectral Tools for Dynamic Tonality and Audio Morphing

William A. Sethares
Andrew J. Milne
Stefan Tiedje

Computer Music Journal, Volume 33, Number 2, Summer 2009, pp. 71-84 (Article)

Published by The MIT Press

William A. Sethares,[*] Andrew J. Milne,[†] Stefan Tiedje,[**] Anthony Prechtl,[†] and James Plamondon[††]

[*]Department of Electrical
and Computer Engineering
University of Wisconsin-Madison
Madison, Wisconsin 53706 USA
sethares@ece.wisc.edu
[†]Department of Music
University of Jyväskylä
P.O. Box 35, 40014 Finland
andymilne@tonalcentre.org
aprechtl@gmail.com
[**]CCMIX
41 Rue des Noyers
93230 Romainville, France
Stefan-Tiedje@gmx.de
[††]Thumtronics
6911 Thistle Hill Way
Austin, Texas 78754 USA
jim@thumtronics.com

Spectral Tools for Dynamic Tonality and Audio Morphing

The Spectral Toolbox is a suite of analysis–resynthesis programs that locate relevant partials of a sound and allow them to be resynthesized at specified frequencies. This enables a variety of routines including *spectral mappings* (changing all partials of a source sound to fixed destination frequencies), *spectral morphing* (continuously interpolating between the partials of a source sound and those of a destination sound), and what we call *Dynamic Tonality* (a novel way of organizing the relationship between a family of tunings and a set of related spectra). A complete application called the TransFormSynth provides a concrete implementation of Dynamic Tonality.

Wendy Carlos looked forward to the day when it would be possible to perform any sound in any tuning: "[N]ot only can we have any possible timbre but these can be played in any possible tuning . . . that might tickle our ears" (Carlos 1987b). The Spectral Toolbox and TransFormSynth address two needs that have previously hindered the realization of this goal: (1) the ability to specify and implement detailed control over the sound's spectrum and timbre and (2) a way to organize the presentation and physical interface of the infinitely many possible tunings.

Computer Music Journal, 33:2, pp. 71–84, Summer 2009

The analysis–resynthesis process at the heart of the Spectral Toolbox is a descendent of the Phase Vocoder (PV) (Moorer 1976; Dolson 1986). But where the PV is generally useful for time stretching (and transposition after a resampling operation), the spectral resynthesis routine SpT.ReSynthesis allows arbitrarily specified manipulations of the spectrum. This is closely related to the fixed-destination spectral manipulations of Lyon (2004) and the audio effects in Laroche and Dolson (1999), but it includes more flexible mappings and an effective decomposition of tonal from noise material. It is also closely related to the spectral-mapping technique of Sethares (1998) but can function continuously (over time) rather than being restricted to a single slice of time. In the simplest application, SpT.Sieve, the partials of a sound (or all the partials in a performance) can be remapped to a fixed template; for example, the partials of a cymbal can be made harmonic, or all partials of a piano performance can be mapped to the scale steps of N-tone equal temperament (i.e., a division of the octave into N equal parts). By specifying the rate at which the partials may change, the spectrum of a source sound can be transformed over time into the spectrum of a chosen destination sound, as demonstrated in the routine SpT.MorphOnBang. Neither the source nor the destination need be fixed. The mapping can be dynamically specified so that a source with partials at frequencies f_0, f_1, $f_2, \ldots, f_n$ is mapped to $g_0, g_1, g_2, \ldots, g_n$. For example, the SpT.Ntet routine can be used to generate sounds with spectra that align with scale steps of the *N*-tone equal tempered scale.

Carlos (1987a) observed that "the timbre of an instrument strongly affects what tuning and scale sound best on that instrument." The most complex of the routines, the TransFormSynth, allows the tuning to be changed dynamically over a broad continuum, and the tuning and timbre to be coupled (to an arbitrary degree). It does this by analyzing existing samples, and then resynthesizing the partials using the routines of Dynamic Tonality. Dynamic Tonality builds upon the concepts of transpositional invariance (Keislar 1987), tuning invariance (Milne, Sethares, and Plamondon 2007, 2008), and dynamic tuning (Milne, Sethares, and Plamondon 2007) by additionally including various just-intonation tunings and allowing the spectrum (overtones) of every note to be dynamically tempered toward a tuning that minimizes sensory dissonance (for intervals tuned to low integer ratios, or temperings thereof).

Transpositional invariance means that a pitch pattern, such as a chord or melody, has the same shape and the same fingering in all musical keys—it requires a controller with a two-dimensional lattice (array) of keys or buttons. *Tuning invariance* means that a pitch pattern has the same shape and the same fingering over a continuum of tunings, even though its intonation may change somewhat. *Dynamic tuning* is the ability to smoothly move between different tunings within such a tuning continuum. Control of the additional parameters required by Dynamic Tonality is facilitated by the Tone Diamond—a novel GUI object that allows the note-tuning and associated spectral-tuning space to be traversed in a simple and elegant fashion.

The TransFormSynth is compatible with conventional controllers (such as a MIDI keyboard or piano roll sequencer), but the transpositional and tuning invariant properties inherent to Dynamic Tonality are best demonstrated when using a compatible two-dimensional controller (such as a MIDI guitar fretboard, a computer keyboard, the forthcoming Thummer controller (www.thummer.com), or the forthcoming software sequencer called Hex (available on the Dynamic Tonality Web site at www.dynamictonality.com).

The TransFormSynth can produce a rich variety of sounds based on (but not necessarily sonically similar to) existing samples, and it provides a straightforward interface for the integrated control of tuning and spectrum. This means that when it is coupled with a two-dimensional controller, it provides an excellent showcase for the creative possibilities opened up by transpositional and tuning invariance and the spectral manipulations of Dynamic Tonality.

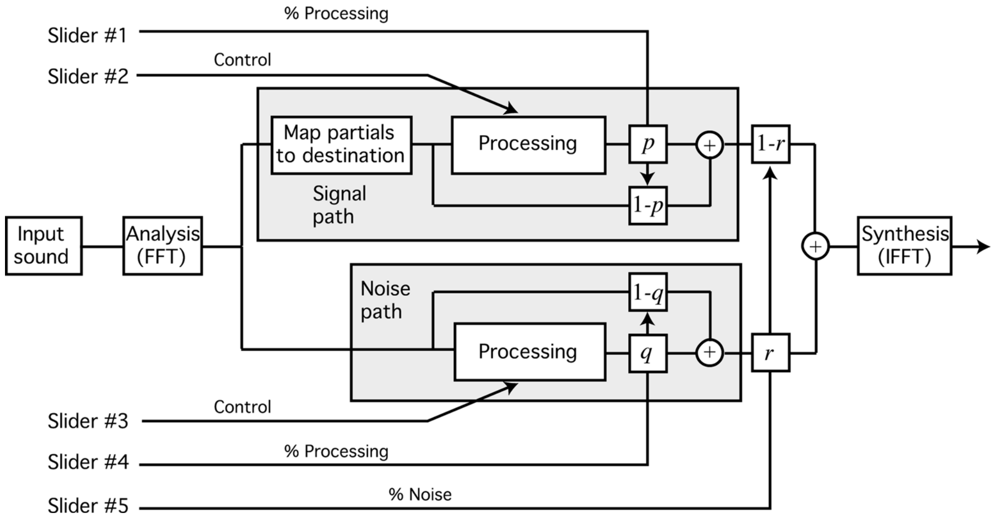
A current version of the Spectral Toolbox (including all the routines mentioned previously) can be downloaded from the Spectral Tools home page at eceserv0.ece.wisc.edu/~sethares/spectoolsCMJ.html. It runs on Windows and Mac OS X using Max/MSP from Cycling '74, including both the full version and the free Max/MSP Runtime version. The spectral-manipulation routines are written in Java, and all programs and source code are released under a Creative Commons license.

Analysis–Resynthesis

To individually manipulate the partials of a sound, it is necessary to locate them. The Spectral Toolbox begins by separating the "signal" (which refers here to the deterministic portion, the most prominent tonal material) from the "noise" (the stochastic portion, consisting of rapid transients or other components that are distributed over a wide range of frequencies). This allows the peaks in the spectrum to be treated differently from the wide-band components. This separation (Serra and Smith 1990) helps preserve the integrity of the tonal material and helps preserve valuable impulsive information such as the attacks of notes that otherwise may be lost due to transient smearing (Laroche and Dolson 1999). The noise parts may be inverted without modification even as the tonal components are changed significantly. The basic flow of information in all of the routines is shown in Figure 1.

We propose a possibly novel technique using a median filter of length *n* that takes the median of each successive set of *n* values. (If a list of numbers

Figure 1. The input sound is broken into frames and then analyzed by a series of overlapping FFTs. The partials (the peaks of the spectrum) follow the top path; they are mapped to their destination frequencies, then optionally processed in the frequency domain. Similarly, the noise portion follows the bottom path and can be processed in the frequency domain before adding and returning to the time domain.



is ordered from smallest to largest, the median is the number in the middle position of the list. For example, the median of [1, 2, 3, 100, 200] is 3.) The noise floor is approximated as the output of a median filter applied to the magnitude spectrum. Because peaks are rarely more than a handful of frequency bins wide, a median filter with length between $m_L = 20$ to $m_L = 40$ allows good rejection of the highs as well as good rejection of the nulls. For example, the left-hand plot in Figure 2 shows the spectrum in a single 4,096-sample frame from Joplin's *Maple Leaf Rag*. The median filter, of length 35, provides a convincing approximation to the noise floor.

For example, SpT.AnalySynth, whose help file is shown in Figure 3, can be used to demonstrate the separation of signal and noise. A sound file is chosen by clicking on the "open" box; alternatively, it is possible to use a live audio input. Pressing "x" starts the processing and displays the magnitudes of the signal and the noise. (The scales are controlled by several boxes that are not shown in Figure 3.) When the "% noise" parameter is set to 0.5, the signal and noise are balanced, and the output resynthesizes the input. With "% noise" set to zero, the output consists of only the signal path. When it is set to 1.0, the output consists of only the noise path. Experimenting with different values of the "threshold multiplier" (which multiplies the noise floor by this factor) and the "maximum # peaks" parameter affects how well the noise-signal separation is accomplished. For example, applying the values shown in the figure to a version of Scott Joplin's *Maple Leaf Rag* gives the *Noisy Leaf Rag* (found on the Spectral Tools home page), where both melody and harmony are removed, leaving only the underlying rhythmic pattern.

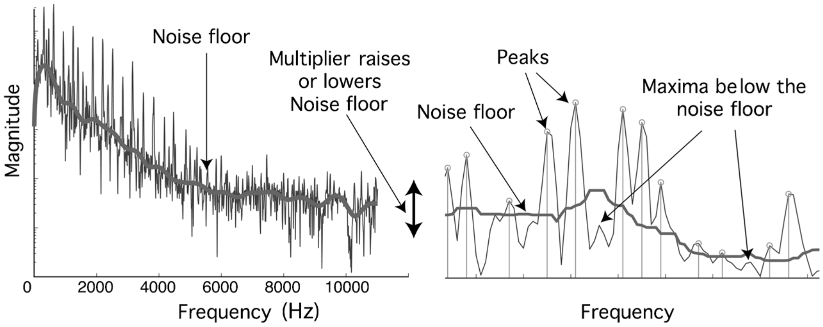
One problem with standard short-time Fourier-transform processing is that the frequencies specified by the Fast Fourier Transform (FFT) are quantized to $\frac{s}{w}$, where s is the sampling rate and w is the size of the FFT window. The phase values from consecutive FFT frames can be used to refine the frequency values of the partials as is often done in the PV (Moorer 1976; Laroche and Dolson 1999).

Suppose that two consecutive frames j and $j+1$ separated by dt seconds have a common peak in the ith frequency bin of the magnitude spectrum (corresponding to a frequency of $i\frac{s}{w}$). Let θ_i^j and θ_i^{j+1} be the phase values of the ith bins, and define $\Delta\theta_i = \theta_i^j - \theta_i^{j+1}$. The frequency estimate is

$$f_i = \frac{1}{dt}\left[\text{Round}\left(\frac{dt\,s}{w}i - \frac{\Delta\theta_i}{2\pi}\right) + \frac{\Delta\theta_i}{2\pi}\right]. \tag{1}$$

The accuracy of this estimate has been shown to approach that of a maximum-likelihood estimate (the value of the frequency f that maximizes the conditional probability of f given the data) for some choices of parameters (Puckette and Brown 1998). In practice, this improves the frequency estimates significantly.

Figure 2. The left-hand graph shows a typical spectrum and the noise floor (the dark line) as calculated using the output of the median filter (multiplied by a constant). The right-hand figure zooms in on a small portion of that graph. The noise floor is used to distinguish the peaks of the spectrum: The M *largest local maxima above the noise floor (shown by the small circles) are treated as significant partials and processed through the top path of Figure 1; data below the noise floor are processed through the bottom path.*



Similarly, in the resynthesis step, the destination frequencies for the partials can be specified to a much greater accuracy than $\frac{s}{w}$ by adjusting the frequencies of the partials using phase differences in successive frames. To be explicit, suppose that the frequency f_i is to be mapped to some value g. Let k be the closest frequency bin in the FFT vector (i.e., the integer k that minimizes $|k\frac{s}{w} - g|$). The kth bin of the output spectrum at time $j+1$ has magnitude equal to the magnitude of the ith bin of the input spectrum with corresponding phase

$$\theta_k^{j+1} = \theta_k^j + 2\pi\, dt\, g. \qquad (2)$$

Together, Equations 1 and 2 allow a more accurate measurement of the frequencies in the source and destination sounds than is possible with a naive use of the FFT. Similarly, in the resynthesis step, the frequencies of the partials can be specified more precisely. The phase values in Equation 2 also guarantee that the resynthesized partials will be continuous across frame boundaries, reducing the likelihood of discontinuities and clicks.

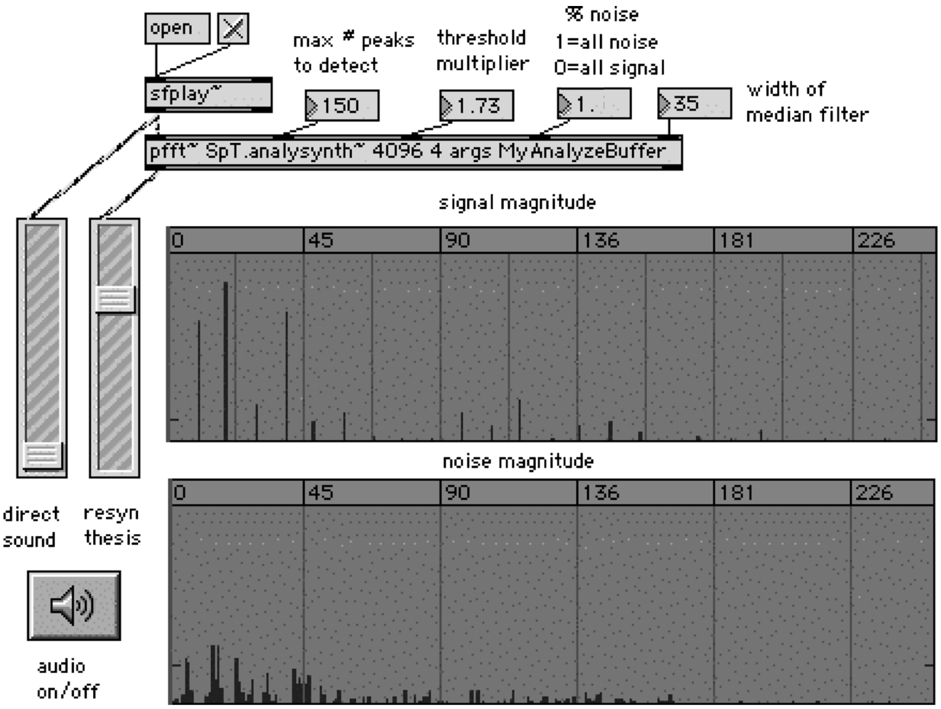
Spectral Mappings

Suppose that a source sound F has n partials $f_1, f_2, \ldots, f_n$ with magnitudes $a_1, a_2, \ldots, a_n$, and let $g_1, g_2, \ldots, g_m$ be m desired partials of the destination sound G. The mapping changes the frequencies of the partials while preserving their magnitudes. Phase values then are created as in Equation 2. A key issue is how to assign the input frequencies f_i to the output frequencies g_j. Two methods that we have found useful are shown schematically in Figure 4. In each diagram, there are two sets of stacked lines that represent the peaks in the magnitude spectra of the source F (on the left) and the destination G (on the right). The arrows show how the assignments are made (and hence which partials of the source map to which partials of the destination). The dark dots represent frequencies that are not in F or G but are nonetheless needed when $n \neq m$.

Multiphonics occur in wind instruments when the coupling between the driver (the reed or lips) and the resonant tube evokes more than a single fundamental frequency. Their sounds tend to be inharmonic and spectrally rich. The two different assignment strategies, described in Figure 4, are contrasted by conducting a 15-sec spectral morph (see "Spectral Morphing," subsequently) between four different pairings of clarinet multiphonics (entitled *multiphonics morph #1*, *#2*, *#3*, and *#4* on the Spectral Tools home page). For each pairing, both assignment strategies—labeled "type 1" for nearest neighbor, and "type 2" for sequential alignment—are used. It can be heard that the different assignment strategies can cause significant differences in the sound, with the partials sometimes appearing to rise, and sometimes to fall.

There are also other ways that the assignments might be made. For example, the sequential alignment might begin with the highest, rather than the lowest, partials. The partials with the maximum magnitudes might be aligned, followed by those with the second largest, and so forth, until all are exhausted. Alternatively, some important pair of partials might be identified (e.g., the largest in magnitude, or the ones nearest the spectral centroid) and the others aligned sequentially above and below.

Figure 3. SpT.AnalySynth demonstrates the decomposition of the magnitude spectrum into signal (the top plot) and noise (the bottom plot). Parameters that affect the decomposition appear across the top. The proportion of signal to noise in the reconstructed signal can be adjusted by changing the "% noise" parameter.



However, early experiments suggest that many of these other methods lead to erratic results in which the pitch changes dramatically in response to small changes in the input sound.

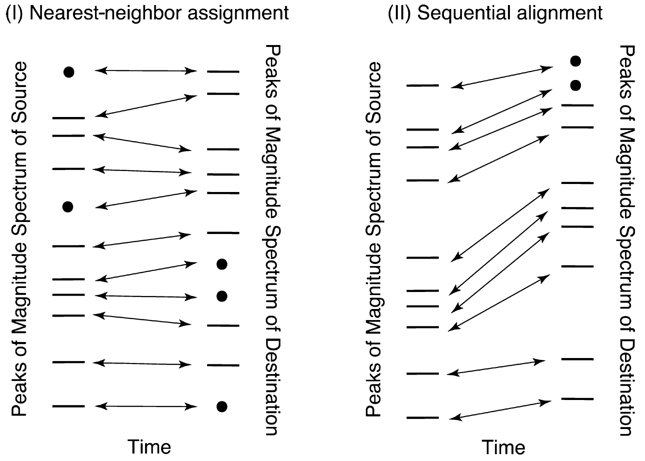
Applications of the Spectral Toolbox

The analysis, spectral mappings, and resynthesis processes described in the previous sections enable a variety of routines including fixed spectral mappings (which transform the partials of a source sound to a fixed set of destination frequencies), spectral morphing (continuously interpolating between the partials of a source sound and a destination), and Dynamic Tonality. These are described in the next few sections.

Fixed Destinations

Perhaps the most straightforward use of the spectral mapping technology is to map the input to a fixed destination spectrum G. For example, since harmonic sounds play an important role in perception, G might be chosen to be a harmonic series built on a fundamental frequency g (i.e., $g_i = ig$) as implemented in the SpT.MakeHarm routine of Figure 5. A sound is played by clicking on "sfplay~" and the root g is chosen either by typing into the rightmost number box or by clicking on the keyboard. (This can easily be replaced with a MIDI input.) The input might be an inharmonic sound such as a gong (see *harmonicgong* on the Spectral Tools homepage), or it may be a full piece such as the *65 Hz Rag* (also found on the Spectral Tools home page) which maps all partials of a performance of Joplin's *Maple Leaf Rag* to integer multiples of $g = 65$ Hz. It is also possible to "play" the mini-keyboard to change the fundamental frequency of the harmonic series over time. *Maple-makeharm* (on the Spectral Tools home page) is a brief improvisation where the fundamental is changed as the piece progresses. One fascinating aspect is that there is a smooth transition from rhythmic pulsation (when the piece is mapped to all harmonics of a low fundamental) to melody (when mapped to all harmonics of a high fundamental).

Figure 4. Two ways to align the partials of the source spectrum with the partials of the destination spectrum. The nearest-neighbor assignment locates $\mathrm{f_i}$ *that are close to* $\mathrm{g_j}$*, and these neighbors are paired. Zero-amplitude partials are added to the list of destination partials and assigned to* $\mathrm{f_i}$ *whenever there are no nearby partials* $\mathrm{g_j}$*, and these are spaced between the nearest destination partials. Similarly, unassigned destination partials are mapped to partials of zero magnitude that are located midway between nearby source partials. The sequential-alignment method assigns the lowest-frequency peak in* F *to the lowest frequency peak in* G*, then pairs the next-lowest frequencies, and so forth, until all are exhausted.*


(I) Nearest-neighbor assignment
Peaks of Magnitude Spectrum of Source
Peaks of Magnitude Spectrum of Destination
Time
(II) Sequential alignment
Peaks of Magnitude Spectrum of Source
Peaks of Magnitude Spectrum of Destination
Time

Similarly, the SpT.Ntet routine maps all partials of the input sound to scale steps of the *N*-tone equal tempered scale. This can be used to create sounds that are particularly appropriate for use in a given *N*-TET scale (Sethares 2004) or to map a complete performance into an approximation of the "same" piece in *N*-TET. For example, *Maple5tet* (found on the Spectral Tools home page) maps all the partials of Joplin's *Maple Leaf Rag* into a fixed 5-TET template. The more sophisticated *Make Believe Rag* (found on the Spectral Tools home page) transforms the same piece into many different *N*-TETs, using different tuning mappings in a way that is somewhat analogous to the change of chord patterns in a more traditional setting. The most general of the fixed destination routines is SpT.Sieve, which maps the input sound to a collection of partials specified by a user-definable table.

Spectral Morphing

Spectral morphing generates sound that moves smoothly between a source spectrum F and a destination spectrum G over a specified time τ (Slaney, Covell, and Lassiter 1996; Cospito and de Tintis 1998). Suppose that F has partials at $f_i, i = 1, 2, \ldots, k$ with magnitude a_i, and G has partials at $g_i, i = 1, 2, \ldots, k$ with magnitude b_i. The two spectra are assumed to be aligned (using one of the methods of Figure 4) so that both have the same number of entries k. Let N_F and N_G be the noise spectra of F and G. Let λ be 0 at the start of the morph and be 1 at time τ. The morph then defines the spectrum at all intermediate times with log-spaced frequencies

$$h(\lambda) = f_i \left(\frac{g_i}{f_i} \right)^{\lambda}, \qquad (3)$$

and linearly spaced intermediate magnitudes

$$m(\lambda) = (1 - \lambda) a_i + \lambda b_i, \qquad (4)$$

and interpolated noise spectra

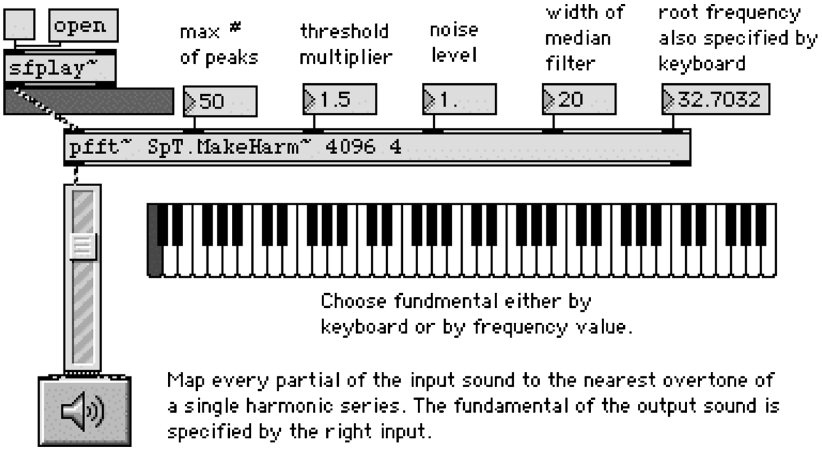
$$N(\lambda) = (1 - \lambda) N_F + \lambda N_G. \qquad (5)$$

Logarithmic interpolation is used in Equation 3 because it preserves the intervallic structure of the partials. The most common example is for harmonic series. If the source and destination each consist of a harmonic series (and if the corresponding elements are mapped to each other in the alignment procedure), then at every λ, the intervening sounds also have a harmonic structure. This is shown mathematically in Appendix A and can be demonstrated concretely using SpT.MorphOnBang, which appears in Figure 6.

To explore the spectral morphing, we recorded Paris-based instrumentalist Carol Robinson playing a number of short clarinet multiphonics whose timbres ranged from soft and mellow to noisy and harsh. Pairs of 1- to 3-second-long multiphonics were spectrally morphed over 15 seconds giving a resulting sound file of about 20 seconds. Examples of these morphs (*multiphonics morph #1*, *#2*, *#3*, and *#4*) were previously referenced in the section "Spectral Mappings."

The next set of examples on the Spectral Tools home page uses a more complex set of morphings with two layers: one layer is a simple quarter-tone melody called *Legend (24-TET melody only)*; the other layer is a succession of multiphonics that have been morphed between each other. Each of the four examples *Legend (melody morphed into multiphonic #1, #2, #3,* and *#4)* uses a different succession of multiphonics, and in each of them the clarinet melody layer is morphed (over 1 second) into the multiphonic layer. Observe that, in these examples, neither the source (the clarinet melody) nor the destination (the multiphonic

Figure 5. SpT.MakeHarm maps all partials of the input sound to a single harmonic series with root specified by the keyboard or by the rightmost number box. The three number boxes labeled "# peaks," "multiplier," and "width of median" set the parameters for spectral peak detection and noise/signal separation. The box labeled "noise level" controls the relative volumes of the signal and noise paths. These parameters are common to all routines in the Spectral Toolbox.



Choose fundmental either by keyboard or by frequency value.

Map every partial of the input sound to the nearest overtone of a single harmonic series. The fundamental of the output sound is specified by the right input.

morph) is a static set of partials, but rather changes as time progresses. The morphing from melody to multiphonics was, if anything, too successful because, although the effect is interesting, the changes to the spectrum of the clarinet render it in many places unrecognizable.

In the piece entitled *Legend of Spectral Phollow*, the process is reversed; rather than choosing the succession of multiphonics in advance, a Max/MSP patch "listens" to the live clarinet melody and reacts by periodically choosing the closest multiphonic and then morphing it into the melody over the same time period (in this example, 20 msec). The quarter-tone melody provides the basis of the melodic material, but the score also calls for significant microtonal improvisation by the performer. The electronics retunes and chooses multiphonics on-the-fly to create an unusual kind of inharmonic backdrop, using the (morphed) multiphonics as an accompaniment analogous to the way block chords may accompany a standard melody. The *Legend of Spectral Phollow* premiered at CCMIX in Paris on 13 July 2006. Carol Robinson played the clarinet, and William Sethares "played" the software.

Dynamic Tonality

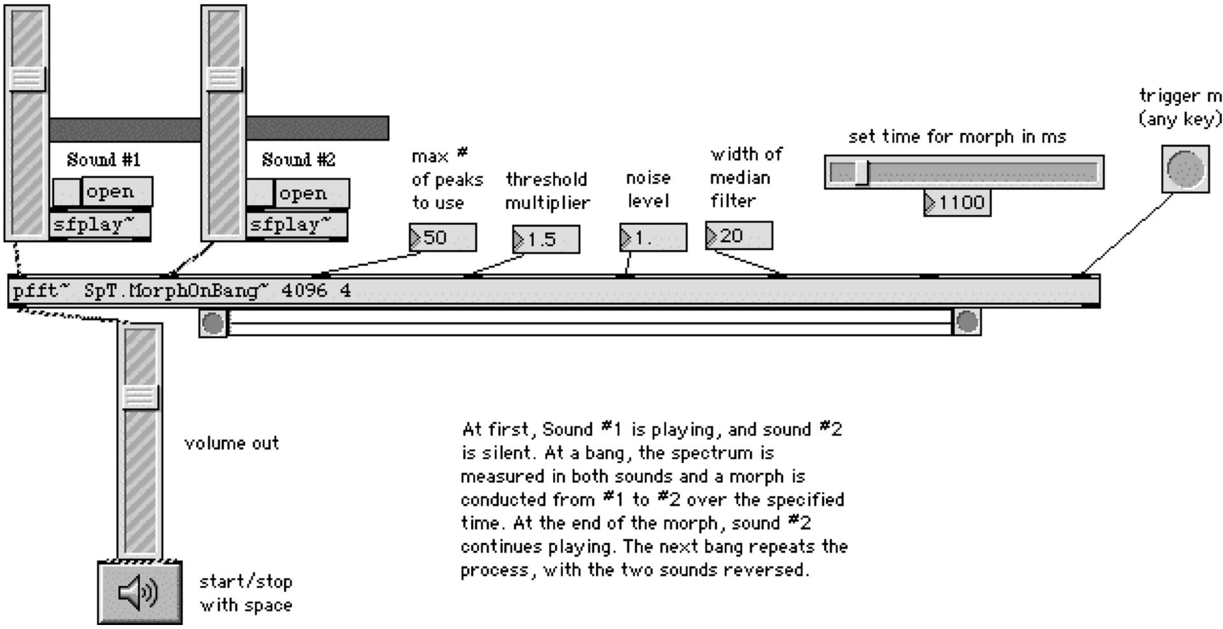
There are many possible tunings: equal temperaments, meantones, circulating temperaments, various forms of just intonation, and so forth. Each seems to require a different method of playing and a different interface, necessitating significant time and effort to master. In Milne, Sethares, and Plamondon (2007, 2008), we introduced a way of parameterizing tunings so that many seemingly unrelated systems can be performed on one keyboard with the same fingerings for the same chords and melodies; this is called tuning invariance. For example, the Syntonic continuum begins at 7-TET, moves through 19-TET, a variety of meantone tunings, 12-TET, 17-TET, 22-TET, and on up to 5-TET (as shown on the main tuning slider in Figure 7). On a musical controller with a two-dimensional array of keys, a chord or melody can usually be fingered the same throughout all the tunings of this continuum.

The TransFormSynth, which is implemented using the same audio routines as described in the Spectral Toolbox, realizes these methods and extends them in two ways. First, the tuning can be moved toward a nearby just intonation. Second, the spectrum of the sound can be tempered along with the tuning. Both of these temperings are implemented using the Tone Diamond—a convenient two-dimensional joystick interface—which is the diamond-shaped object at the top-left of Figure 7.

A *Dynamic Tonality* synthesizer (like TransFormSynth) has a small number of parameters that enable many musically useful, and relatively unexplored, features:

1. The continuous parameters α, β, and γ (explained subsequently) move the tuning between a number of equal temperaments

Figure 6. SpT.MorphOnBang can be applied to individual sounds or to complete musical performances. The time over which the morph occurs is specified by the slider and is triggered by the button on the right.



At first, Sound #1 is playing, and sound #2 is silent. At a bang, the spectrum is measured in both sounds and a morph is conducted from #1 to #2 over the specified time. At the end of the morph, sound #2 continues playing. The next bang repeats the process, with the two sounds reversed.

(e.g., 7-TET, 31-TET, 12-TET, 17-TET, and 5-TET), non-equal temperaments (e.g., quarter-comma meantone, and Pythagorean), circulating temperaments, and closely related just intonations.

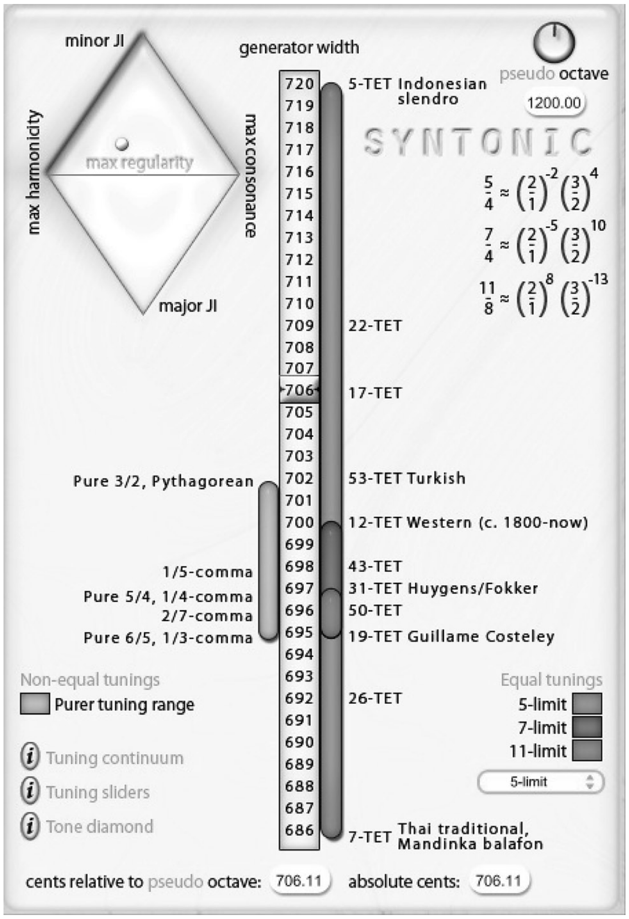
2. The mapping to a two-dimensional lattice of buttons j and k on a musical controller provides the same fingering pattern for all tonal intervals across all possible keys and tunings within any given tuning continuum (Milne, Sethares, and Plamondon 2008).
3. The continuous parameter δ moves the timbre from being perfectly harmonic to being perfectly matched to the tuning, thus minimizing sensory dissonance (Sethares 1993).
4. The discrete parameter c switches between a number of different tuning continua, some of which embed traditional well-formed scales like the pentatonic, diatonic, and chromatic (Carey and Clampitt 1989), and some of which embed radically different well-formed scales (e.g., scales with three large steps and seven small steps per octave).

Each of these parameters is defined and explained in more depth in the following subsections.

Generator Tunings (α and β) and Note Coordinates (j and k)

Invariant fingering over a tuning continuum requires a linear mapping of the notes of a higher-dimensional just intonation to the notes of a one- or two-dimensional temperament (such as 12-TET or quarter-comma meantone), and a linear mapping of these tempered notes to a two-dimensional array of buttons or keys on a musical controller. (The dimensionality of a tuning is equivalent to the minimum number of unique intervals (expressed in $\log(f)$) that are required to generate, by linear combination, all of that tuning's intervals.) Perhaps the simplest way to explain the system is by example. A *p-limit just intonation* contains intervals tuned to ratios that can be factorized by prime numbers up to, but no higher than, p (Partch 1974). Consider 11-limit just intonation (in which $p = 11$), which consists of all the intervals generated by integer multiples of the primes 2, 3, 5, 7, and 11. Thus simple intervals, such as the just fifth or just major third, can be represented as the frequency ratios $\frac{3}{2} = 2^{-1}3^{1}$ and $\frac{5}{4} = 2^{-2}5^{1}$, respectively, while a less simple interval such as the just major seventh (a perfect fifth plus a major third) is $\frac{15}{8} = 2^{-3}3^{1}5^{1}$. A comma

Figure 7. The tuning panel of TransFormSynth displaying the Syntonic tuning continuum. The vertical slider at the center controls the β-tuning (typically, the perfect fifth); the rotary knob at the top-right controls the α–tuning (typically, the octave); the Tone Diamond (at the top left) controls the value of γ (the proximity to a just intonation tuning) and the value of δ (the degree of spectral tempering). See text for further details.



pseudo octave 1200.00

cents relative to pseudo octave: 706.11 absolute cents: 706.11

$(i_1, i_2, i_3, i_4, i_5) \in \mathbb{Z}^5$, in 11-limit just intonation, is a set of integers that tempers (changes the numerical values of) the generators so that $G_1^{i_1} G_2^{i_2} G_3^{i_3} G_4^{i_4} G_5^{i_5} = 1$. For example, the well-known syntonic comma, which can be written in fractional form as $\frac{81}{80}$, is represented by $(-4, 4, -1, 0, 0)$, because it is equal to $2^{-4}3^{4}5^{-1}7^{0}11^{0}$. A system of commas can be represented by a matrix of integer values, so the commas $G_1^{-7}G_2^{-1}G_3^{1}G_4^{1}G_5^{1} = 1, G_1^{1}G_2^{2}G_3^{-3}G_4^{1}G_5^{0} = 1$, $G_1^{-4}G_2^{4}G_3^{-1}G_4^{0}G_5^{0} = 1$, can be represented as the matrix $\mathbf{C} = \begin{pmatrix} -7 & -1 & 1 & 1 & 1 \\ 1 & 2 & -3 & 1 & 0 \\ -4 & 4 & -1 & 0 & 0 \end{pmatrix}$, which has a null space (kernel) $\mathcal{N}(\mathbf{C}) = \begin{pmatrix} 13 & 24 & 44 & 71 & 0 \\ 10 & 13 & 12 & 0 & 71 \end{pmatrix}^{\mathrm{T}}$, where $(\bullet)^{\mathrm{T}}$ is the transpose operator. This matrix is transposed and then written in row-reduced echelon form to give the transformation matrix $\mathbf{R} = \begin{pmatrix} 1 & 0 & -4 & -13 & 24 \\ 0 & 1 & 4 & 10 & -13 \end{pmatrix}$. Using $\mathbf{R} * (i_1, i_2, i_3, i_4, i_5)^{\mathrm{T}} = (j, k)^{\mathrm{T}}$, the matrix $\mathbf{R}$ transforms any interval $(i_1, i_2, i_3, i_4, i_5) \in \mathbb{Z}^5$ into a similarly sized (i.e., tempered) interval $(j, k) \in \mathbb{Z}^2$. A basis (i.e., a set of vectors that can, in linear combination, represent every vector in that space) for the generators can be found by inspection of the columns of $\mathbf{R}$ as $G_1 \mapsto \alpha, G_2 \mapsto \beta, G_3 \mapsto \alpha^{-4}\beta^{4}, G_4 \mapsto \alpha^{-13}\beta^{10}$, and $G_5 \mapsto \alpha^{24}\beta^{-13}$. Thus, every interval in the continuum (this is the 11-limit Syntonic continuum shown in Figure 7) can be represented as integer powers of the two generators α and β—that is, as $\alpha^j \beta^k$. For further information and examples, see Milne, Sethares, and Plamondon (2008).

This means that if α and β are mapped to a basis $(\boldsymbol{\psi}, \boldsymbol{\omega})$ of a button lattice (i.e., $\alpha^j \beta^k \mapsto j\boldsymbol{\psi} + k\boldsymbol{\omega}$), such as the Thummer's (see Figure 8), then the fundamental frequency of any button of coordinate (j, k) with respect to that basis, is given by

$$f_{j,k}(\alpha, \beta, f_{\mathrm{r}}) = f_{\mathrm{r}} * \alpha^j \beta^k, \tag{6}$$

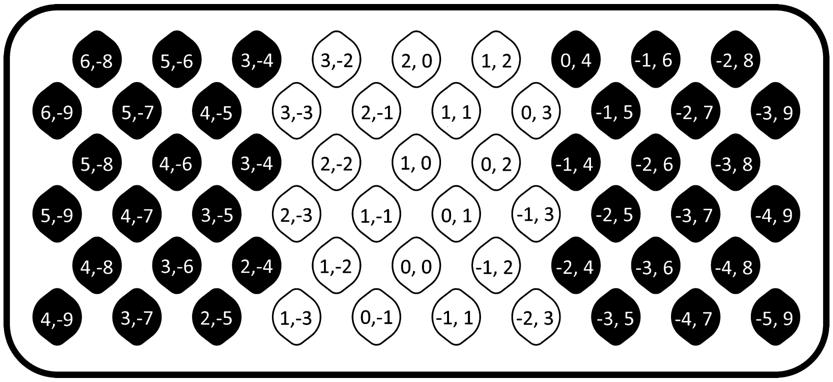
where f_{r} is the frequency of the reference note. By default the *reference note* (0, 0), on which the pitches of all other notes are based, is D3 (whose concert pitch is 146.83 Hz).

In the Syntonic continuum, the value of α is near 2/1 and can be adjusted by the rotary knob labeled "pseudo octave" at the top-right of Figure 7; the value of β is near 3/2 and is specified (in cents) by the main tuning slider. Altering the β-tuning while playing allows a keyboard performer to emulate the dynamic tuning of string and aerophone players who prefer Pythagorean (or higher) tunings when playing expressive melodies, quarter-comma meantone when playing consonant harmonies, and 12-TET when playing with fixed pitch instruments such as the piano (Sundberg, Friberg, and Frydén 1989).

Related Just Intonations (γ)

The vertical dimension of the Tone Diamond (at the top-left of Figure 7) alters the tuning in a different way—by moving it towards a related 5-limit just intonation. Just intonations contain many intervals tuned to ratios of small integers (3/2, 4/3, 5/4, 6/5, 7/5, 7/6, etc.), and these intervals are typically

*Figure 8. The coordinates (*j, k*) of the Thummer's button lattice, when using its default Wicki note layout (Milne, Sethares, and Plamondon 2008).*



thought to be maximally consonant and "in tune" when using sounds with harmonic spectra. For this reason, just intonation has been frequently cited as an ideal tuning (e.g., by Helmholtz 1954; Partch 1974; Mathieu 1997). However, 5-limit just intonation (JI) is three-dimensional, and higher-limit JI's have even more dimensions, making it all but impossible to avoid "wolf" intervals when mapping to a fixed pitch instrument (Milne, Sethares, and Plamondon 2007).

Deciding precisely which JI ratios should be used also presents a problem, because there is always ambiguity about precisely which JI interval is represented by a tempered interval (because the mapping matrix R is many-to-one, any "reverse-mapping" is somewhat ambiguous). For this reason we provide two aesthetically motivated choices: "Major JI," at the bottom of the diamond, maximizes the number of justly tuned major triads (of ratio 4:5:6); and "Minor JI," at the top of the diamond, maximizes the number of justly tuned minor triads (of ratio 10:12:15).

The major and minor JI tuning ratios (relative to the reference note) for every note (j, k) are stored in a table. The major JI values are used when the control dot is in the lower half of the Tone Diamond (i.e., $\mathrm{sgn}(\gamma) = -1$), the minor JI values are used when the control dot is in the upper half of the Tone Diamond (i.e., $\mathrm{sgn}(\gamma) = 1$). Every different tuning continuum c requires a different set of values. The vertical dimension of the Tone Diamond controls how much the tuning is moved towards these JI values, denoted $p_{c,\mathrm{sgn}(\gamma),j,k}$, using the formula $(\frac{p_{c,\mathrm{sgn}(\gamma),j,k}}{\alpha^j\beta^k})^{|\gamma|}$, where $-1 \le \gamma \le 1$ is the position of the control dot on the Tone Diamond's y-axis. This means the frequency of any note can be calculated accordingly as

$$f_{j,k}(\alpha, \beta, f_\mathrm{r}, c, \gamma) = f_\mathrm{r} * \alpha^j \beta^k * \left(\frac{p_{c,\mathrm{sgn}(\gamma),j,k}}{\alpha^j \beta^k}\right)^{|\gamma|}. \quad (7)$$

The Tone Diamond and main tuning slider, therefore, facilitate dynamic tuning changes between many different tuning systems. When the Tone Diamond's control point is anywhere along the central horizontal line (the "Max. Regularity" line), the tuning is a one- or two-dimensional tuning such as 12-TET or quarter-comma meantone, as shown on the main tuning slider. When the control point is moved upward or downward the tuning moves towards a related just intonation. The tunings that are intermediate between perfect regularity and JI are like the circulating temperaments of Kirnberger and Vallotti in that every key has a (slightly) different tuning. And all of these tunings have the same fingering when played on a 2D lattice controller.

Spectral Tempering (δ)

The resynthesis method employed by the TransFormSynth enables the tuning of every partial to be independently adjusted in real time. To make these dynamic alterations of spectral tuning musically useful, a Dynamic Tonality synthesizer can tune the partials to be perfectly harmonic (i.e., with partials whose frequencies are at integer multiples), temper them to match the current tuning (so that the partials coincide with other notes found in the scale), or tune them anywhere in between. For any given value of γ, the degree of coupling between spectrum and tuning can be dynamically changed by moving the Tone Diamond's control dot left or right—when it is on the left edge of the diamond, the sound is perfectly harmonic; when it is on the right edge, the sound is perfectly matched to the current tuning. To implement this, it is necessary to define a method by which a spectrum can be matched to a tuning.

The matrix **R** can be used to parameterize the timbres so as to minimize sensory dissonance when playing in the "related" scale (Sethares 1993). Partials of a harmonic (or approximately harmonic) sound are indexed by integers and can be represented as a vector in $\mathbb{Z}^5$. Thus $2 \leftrightarrow (1, 0, 0, 0, 0)^{\mathrm{T}} \equiv \boldsymbol{h}_2$, $3 \leftrightarrow (0, 1, 0, 0, 0)^{\mathrm{T}} \equiv \boldsymbol{h}_3$, $4 \leftrightarrow (2, 0, 0, 0, 0)^{\mathrm{T}} \equiv \boldsymbol{h}_4$, $5 \leftrightarrow (0, 0, 1, 0, 0)^{\mathrm{T}} \equiv \boldsymbol{h}_5$, $6 \leftrightarrow (1, 1, 0, 0, 0)^T \equiv \boldsymbol{h}_6$, etc. Every different tuning continuum c (such as the Syntonic, discussed previously) has a different **R** matrix, and these $\mathbf{R}_c$ are stored in a table. The ith partial can therefore be tempered to $\mathbf{R}_c\boldsymbol{h}_i = \binom{m_{c,i}}{n_{c,i}}$ and then mapped to $\alpha^{m_{c,i}}\beta^{n_{c,i}}$. Thus the timbre is tempered in a fashion consistent with (and using the same interface as) the tuning. It is easy to verify that these temperings are the same as those identified by Sethares (2004) for the special case of equal temperaments.

Both the horizontal and vertical dimensions of the Tone Diamond control how much of this tempering is applied using the interpolation formula $i(\frac{\alpha^{m_{c,i}}\beta^{n_{c,i}}}{i})^{\delta}$, where $\delta = x - \frac{|\gamma|}{2}$, $0 \le x \le 1$ is the position of the control dot on the Tone Diamond's x-axis, and γ is the position on the y-axis. This means that when the control dot is anywhere on either of the two line segments at the diamond's left boundary (labeled "max harmonicity"), then $\delta = 0$, and the sound remains harmonic with integer partials i. When the control dot is fully to the right, $\delta = 1$, and the partials are tempered to $\alpha^{m_{c,i}}\beta^{n_{c,i}}$; whenever the control dot is on either of the two line segments at the diamond's right boundary (labeled "max consonance"), the partials are always fully related to the tuning. It also means that if the control dot is moved vertically upward from the center of the diamond, not only is the tuning modified, but also the spectral tempering δ (from 0.5 in the center to 0 at the top or bottom end).

The frequency of any partial can, therefore, be defined in terms of $\alpha, \beta, j, k, c, i, \gamma$, and δ, using the following formula:

$$\begin{aligned} f_{i,j,k}(\alpha, \beta, f_{\mathrm{r}}, c, \gamma, \delta) &= f_{\mathrm{r}} * \alpha^{j}\beta^{k} * \left(\frac{p_{c,\mathrm{sgn}(\gamma),j,k}}{\alpha^{j}\beta^{k}}\right)^{|\gamma|} \\ &\quad * i\left(\frac{\alpha^{m_{c,i}}\beta^{n_{c,i}}}{i}\right)^{\delta} \\ &= f_{\mathrm{r}} * (\alpha^{j}\beta^{k})^{1-|\gamma|} * p_{c,\mathrm{sgn}(\gamma),j,k}^{|\gamma|} * i^{1-\delta} \\ &\quad * (\alpha^{m_{c,i}}\beta^{n_{c,i}})^{\delta}. \end{aligned} \tag{8}$$

If α, β, and $p_{c,\mathrm{sgn}(\gamma),j,k}$, are expressed in cents, which may be more convenient for the user, this formula can be rewritten as

$$\begin{aligned} &f_{i,j,k}(\alpha, \beta, f_{\mathrm{r}}, c, \gamma, \delta) \\ &\quad = f_{\mathrm{r}} * 2^{\frac{(1-|\gamma|)(j\alpha+k\beta)+|\gamma|p_{c,\mathrm{sgn}(\gamma),j,k}}{1200}} * i^{1-\delta} \\ &\quad * 2^{\frac{\delta(\alpha m_{c,i}+\beta n_{c,i})}{1200}}. \end{aligned} \tag{9}$$

The Tone Diamond is labeled to show that the further the control point is from the "max harmonicity" line, the less harmonic its partials; the further the control point is from the "max consonance" line, the less related its partials are to the tuning; the further the control point is from the "max regularity" line, the less regular are its interval sizes. The diamond clearly illustrates how every possible position of the control point represents a compromise between maximal harmonicity, maximal consonance, and maximal regularity; no system can have all three at the same time. (Any corner of the diamond provides two out of the three.)

Tuning Continua (c) and Compositional Possibilities

Although this article has focused on the Syntonic tuning continuum, there are numerous other useful continua, each with unique and unfamiliar intervallic structures. The TransFormSynth currently implements two other continua—"Magic" and

"Hanson" (Erlich 2006)—which open up interesting compositional avenues. They contain scales that embed numerous major and minor triads, but these scales have a radically different structure than those found in any standard Western tuning. For example, the Magic continuum has a ten-note well-formed scale (with seven small steps and three large steps) that contains ten major or minor triads; the Hanson continuum has an eleven-note well-formed scale (with seven small steps and four large steps) that also contains ten major or minor triads. *Magic Traveller* (found on the Spectral Tools home page) uses this Magic scale. It may well be that the chords in these systems have functional relationships that are quite different from those found in standard diatonic/chromatic harmony. Such systems, therefore, open up the possibility of an aesthetic research program similar to that which may be said to have characterized the development of common-practice music from the rise of functional harmony in the seventeenth century to the "crisis of tonality" at the end of the nineteenth.

But the well-structured tonal relationships found in these continua do not support only a strictly tonal compositional style. Serial (and other "atonal") compositional techniques are just as applicable to these alternative continua, as are techniques that explore the implications of unusual timbral combinations and structures. Each continuum offers a unique set of mathematical possibilities and constraints. For example, the familiar 12-note division of the octave includes many factors (2, 3, 4, and 6), thus enabling interval classes of these sizes to cycle back to the starting note, and modes of limited transposition to be formed (Messiaen 1944). Conversely, a 13-note division of the octave, which can be made to sound quite "in-tune" when the spectrum is tempered to the Magic continuum, has no factors and so contains no modes of limited transposition and no interval cycles. The 15-note division found in Hanson has factors of 3 and 5, suggesting a quite different set of structural possibilities. *ChatterBar* and *Lighthouse* (found on the Spectral Tools homepage) are both non-serial "atonal" pieces—in 53-TET Syntonic and 11-TET Hanson, respectively.

Alongside these structural possibilities are the dynamic variations in tuning and timbre that can be easily controlled (and even notated) with the α, β, γ, and δ parameters. Smooth changes of tuning and timbre are at the core of *C2ShiningC*, while in *Shred* (found on the Spectral Tools home page), the music switches from 12-TET to 5-TET Syntonic. We believe Dynamic tonality offers a rich set of compositional possibilities of both depth and simplicity.

Discussion

The analysis–resynthesis method used by the Spectral Toolbox allows the independent control of both frequency and amplitude for every partial in a given sound. However, because a typical musical sound consists of tens or even hundreds of audible partials, it is apparent that their individual manipulation is not necessarily practical. To reduce information load and retain musical relevance, there is need for an organizational routine that parameterizes spectral information in a simple and musically meaningful interface. The Spectral Toolbox has addressed this problem by providing three different routines: (1) mapping partials to a fixed destination, (2) morphing between different spectra, and (3) Dynamic Tonality.

Although we have so far only discussed the reconstruction of preexisting sounds, it is also possible to manipulate the harmonic information of purely synthesized sound. The ideas presented in this article are applicable to virtually any synthesis method that allows complete control over harmonic information. For example, The Viking (Milne and Prechtl 2008) is an additive synthesizer that implements Dynamic Tonality in the same manner as the TransFormSynth, except that it synthesizes each partial with its own sinusoidal oscillator. Similarly, 2032 (available on the Dynamic Tonality Web site) is a modal synthesizer that implements Dynamic Tonality in a physical-modeling algorithm. In this case, an excitation signal is fed through a series of resonant filters that represent specific partials through their individual feedback coefficients.

There are benefits pertaining to each of these synthesis methods: Additive synthesis is, relatively speaking, computationally efficient, whereas modal synthesis, at the cost of greater computational power, enables realistic and dynamic physical modeling. However, the analysis–resynthesis method is interesting because it enables the harmonic manipulation of any sound, and it can do so for both fixed and live audio inputs. This means that, given its relatively simple user interface, the Spectral Toolbox has the capacity to provide novel and worthwhile approaches to computer-music composition and performance. The musical examples available on the Web site are intended to illustrate some of these artistic possibilities.

Beyond the artistic benefits described herein, this work also provides strong implications for music research, particularly in the area of cognition. The mutual control of tuning and timbre facilitates a deeper examination of the musical ramifications that such a relationship entails. Perhaps of greatest interest is how formerly inaccessible (that is, in an aesthetic sense) tunings can be rendered accessible through the timbral manipulations described in this article. Such an idea calls for further research regarding varying forms of dissonance—most notably melodic dissonance (Van der Merwe 1992; Weisethaunet 2001)—and harmonic tonality in general. Such concepts can now take alternative tunings into account. Tools such as the Spectral Toolbox can facilitate a wider use of microtonality in electroacoustic composition, performance, and research.

Acknowledgments

The authors would like to thank Gerard Pape for helpful discussions and Carol Robinson for the excellent performance of *Legend*.

References

Carey, N., and D. Clampitt. 1989. "Aspects of Well-Formed Scales." *Music Theory Spectrum* 11:187–206.

Carlos, W. 1987a. "Tuning: At the Crossroads." *Computer Music Journal* 11(1):29–43.

Carlos, W. 1987b. *Secrets of Synthesis*. Audio compact disc. East Side Digital ESD 81702.

Cospito, G., and R. de Tintis. 1998. "Morph: Timbre Hybridization Tools Based on Frequency Domain Processing." Paper presented at the 1998 Workshop on Digital Audio Effects (DAFx-98), Barcelona, 21 November.

Dolson, M. 1986. "The Phase Vocoder: A Tutorial." *Computer Music Journal* 10(4):14–27.

Erlich, P. 2006. "A Middle Path Between Just Intonation and the Equal Temperaments, Part 1." *Xenharmonikon* 18:159–199.

Helmholtz, H. L. F. 1954. *On the Sensations of Tone*. New York: Dover.

Keislar, D. 1987. "History and Principles of Microtonal Keyboards." *Computer Music Journal* 11(1):18–28. Published with corrections in 1988 as "History and Principles of Microtonal Keyboard Design," Report STAN-M-45, Stanford University. Available online at ccrma.stanford.edu/STANM/stanms/stanm45/. (Accessed 30 December 2008.)

Laroche, J., and M. Dolson. 1999. "Improved Phase Vocoder Time-Scale Modification of Audio." *IEEE Transactions on Audio and Speech Processing* 7(3): 323–332.

Lyon, E. 2004. "Spectral Tuning." *Proceedings of the 2004 International Computer Music Conference*. San Francisco: International Computer Music Association, pp. 375–377.

Mathieu, W. A. 1997. *Harmonic Experience*. Rochester: Inner Traditions.

Messiaen, O. 1944. *Technique de mon langage musical*. Paris: Leduc.

Milne, A. J., and A. Prechtl. 2008. "New Tonalities with the Thummer and The Viking." *Paper presented at the Third International Haptic and Auditory Interaction Design Workshop.* Jyväskylä, Finland: University of Jyväskylä, pp. 20–22.

Milne, A. J., W. A. Sethares, and J. Plamondon. 2007. "Isomorphic Controllers and Dynamic Tuning: Invariant Fingering over a Tuning Continuum." *Computer Music Journal* 31(4):15–32.

Milne, A. J., W. A. Sethares, and J. Plamondon. 2008. "Tuning Continua and Keyboard Layouts." *Journal of Mathematics and Music* 2(1):1–19.

Moorer, J. A. 1976. "The Use of the Phase Vocoder in Computer Music Applications." *Journal of the Audio Engineering Society* 26(1/2):42–45.

Partch, H. 1974. *Genesis of a Music*. New York: Da Capo Press.

Puckette, M. S., and J. C. Brown. 1998. "Accuracy of Frequency Estimates Using the Phase Vocoder." *IEEE Transactions on Speech and Audio Processing* 6(2):166–176.

Serra, X., and J. Smith. 1990. "Spectral Modeling Synthesis: A Sound Analysis/Synthesis System Based on a Deterministic Plus Stochastic Decomposition." *Computer Music Journal* 14(4):12–24.

Sethares, W. A. 1993. "Local Consonance and the Relationship between Timbre and Scale." *Journal of the Acoustical Society of America* 94(3):1218–1228.

Sethares, W. A. 1998. "Consonance-Based Spectral Mappings." *Computer Music Journal* 22(1):56–72.

Sethares, W. A. 2004. *Tuning, Timbre, Spectrum, Scale*, 2nd ed. London: Springer-Verlag.

Slaney, M., M. Covell, and B. Lassiter. 1996. "Automatic Audio Morphing." *Proceedings of the 1996 ICASSP*, Atlanta, GA.

Sundberg, J., A. Friberg, and L. Frydén. 1989. "Rules for Automated Performance of Ensemble Music." *Contemporary Music Review* 3(1):89–109.

Van der Merwe, P. 1992. *Origins of the Popular Style*. Oxford: Clarendon Press.

Weisethaunet, H. 2001. "Is There Such a Thing as the 'Blue Note'?" *Popular Music* 20(1):99–116.

Appendix: Preservation of Intervallic Structure Under Logarithmic Interpolation

Suppose that the n source peaks f_i and the n destination peaks g_i have the same intervallic structure, i.e., that

$$\frac{f_{i+1}}{f_i} = \frac{g_{i+1}}{g_i} \tag{10}$$

for $i = 1, 2, \ldots, n-1$. Morphing the two sounds using the logarithmic method (Equation 3) creates a collection of intermediate sounds with peaks at

$$h_i(\lambda) = f_i \left(\frac{g_i}{f_i}\right)^{\lambda}. \tag{11}$$

Then, for every $0 \le \lambda \le 1$, the intervallic structure in the $h_i(\lambda)$ is the same as that in the source and destination. To see this, observe that

$$\frac{h_{i+1}(\lambda)}{h_i(\lambda)} = \frac{f_{i+1}\left(\frac{g_{i+1}}{f_{i+1}}\right)^{\lambda}}{f_i\left(\frac{g_i}{f_i}\right)^{\lambda}} = \frac{f_{i+1}}{f_i}\left(\frac{g_{i+1}}{f_{i+1}}\right)^{\lambda}\left(\frac{f_i}{g_i}\right)^{\lambda}$$

$$= \frac{f_{i+1}}{f_i}\left(\frac{g_{i+1}}{g_i}\frac{f_i}{f_{i+1}}\right)^{\lambda} = \frac{f_{i+1}}{f_i} \tag{12}$$

The last equality follows directly from Equation 10. In particular, if the f_i and g_i are the n partials of harmonic sounds (though perhaps with different magnitudes and different fundamentals) then the interpolated sounds $h(\lambda)$ are also harmonic, with spectra that smoothly connect f and g and with fundamental frequency (and hence, most likely, with pitch) that moves smoothly from that of f to that of g.