QUICKMQ: A SOFTWARE TOOL FOR THE MODIFICATION OF TIME-VARYING SPECTRUM ANALYSIS FILES

Stephen William Berkley
BIAS
email: sberkley@crl.com, berkley@onyx.dartmouth.edu

ABSTRACT
The QuickMQ application for the Macintosh computer provides numerous frequency-domain transformations using Lemur analysis files. These transformations include convolution, deconvolution, spectrum mix, granular desynthesis, brightening, harmonic rotation, and a unique implementation of a spectrum-processing language based on the graphics manipulation language Popi by Gerard Hoxmann (1988). "Frames" of peaks may be viewed as movies, where the change in spectrum may be observed over time. QuickMQ allows the user to describe frequency-domain translation algorithms files that alter frequency, time, and amplitude to produce new audio transformations from the original analysis files.

INTRODUCTION
QuickMQ belongs to a broad class of synthesis techniques referred to as synthesis by analysis. Techniques of synthesis by analysis were initially investigated in the work of Jean-Claude Risset (1982).

The QuickMQ application modifies and saves "Lemur" MQ files. Lemur (1994), created by Kelly Fitz, Brian Holloway, and Bill Walker at CERL, University of Illinois, creates analysis files by analyzing sound files. Lemur applies a series of windowed STFTs (Short-Time Fourier Transforms), and then picks the most "prominent" peaks using a very crude psychoacoustical approximation of masking. Peaks are stored in the frames, each peak points to its successor peak in the next frame, forming tracks.

The resulting Lemur analysis files may then be processed with the QuickMQ application as either vertical data structures (frame-based transformations) or horizontal data structures (track-based transformations).

Modifying a sound with the QuickMQ application involves: 1. Creating a Lemur analysis file of a soundfile with the Lemur application. 2. Modifying the analysis file with QuickMQ, or creating a new analysis file. 3. Resynthesizing the analysis file to a soundfile with the Lemur application.

QuickMQ executes its transformations as a disk-based processing environment, since Lemur analysis files can grow very large. QuickMQ reads and writes frames and peaks from the source file) and destination file as they are needed. While this allows for the transformation of infinitely large analysis files, it slows computational times due to disk accessing time.

ONE-FILE MODIFICATIONS
The Brightness Variation transformation changes the timbral brightness of a Lemur analysis file. This is accomplished by applying a time-varying brightness ramp, analogous to a high-pass filter, to the analysis file. The ramp may have a "center" frequency at which point the ramp is applying a zero coefficient to the analysis peaks.

The Harmonic Rotation transformation produces thousands of variations from a single analysis file. By "rotating" the amplitude or frequency envelopes of the tracks (partials) in an analysis file, different harmonic components of the analysis file are emphasized. The order of rotation determines the distance of the envelope away from the current envelope to use as a replacement envelope, illustrated below:

<table>
<thead>
<tr>
<th>Source Analysis File</th>
<th>Rotated Analysis File (order=1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partial 1: Envelope 1</td>
<td>Partial 1: Envelope 2</td>
</tr>
<tr>
<td>Partial 2: Envelope 2</td>
<td>Partial 2: Envelope 3</td>
</tr>
<tr>
<td>Partial 3: Envelope 3</td>
<td>Partial 3: Envelope 1</td>
</tr>
</tbody>
</table>

Currently the Harmonic Rotation algorithm interpolates envelopes when replacing 1) a longer envelope with a shorter envelope or 2) replacing a shorter envelope with a longer envelope. Smoothing allows the user to specify an envelope for the attack and the decay of the amplitude envelope for each track. Smoothing becomes necessary when short, loud tracks are imposed onto longer tracks. This process reduces clicking and other artifacts generated by the rotation algorithm.
The user can choose to rotate amplitude envelopes or frequency envelopes. When rotating amplitude envelopes, the frequency envelopes remain fixed; only the frequency envelopes of tracks in the analysis file will be modified. When rotating frequency envelopes, the amplitude envelopes remain fixed while the frequency envelopes are modified.

The Harmonic Rotation algorithm is as follows: 1) Find all of the tracks in the analysis file. 2) Sort the tracks based on user-selected key (track starting frequency, averaged track magnitude, track duration, or start time). 3) Read the envelope of the next track in. 4) Read the envelope of the track current_track + rotate_order away (note: modulo arithmetic allows for wraparound). 5) Rewrite the envelope of the current track with the "destination" envelope, interpolated. 6) Continue at step 3) until finished.

The Harmonic Rotation algorithm provides a large set of variations upon a single sound source for composers. Future enhancements to the Harmonic Rotation algorithm may include the ability to rotate peaks based on windows of frames, rather than tracks, where users enter a rotation order for each window.

The Granular algorithm allows the user to poke holes in an analysis file in a systematic way. The effect is to "granularize" the analysis file by compromising its continuity over time. Granular synthesis is an additive technique of composing sound gestures with hundreds and even thousands of small "granules" composed of simple or complex tones. Granular synthesis has only recently begun to be used as a synthesis technique because of the computationally prohibitive processing time (Roads). QuickMQ provides a method for the subtractive synthesis of sonic events with granular techniques. The aural result usually sounds like the reduction of a sound into droplets of water.

Currently the exact procedure for achieving a "Morph" between two soundfiles has not been adequately determined. Grey and Moore (1977) accomplished transformations of one sound into another through the interpolation of harmonic envelopes of sounds. However, since the spectrum of an analysis file is simply a result of a dynamic system [Balzano], attempting to interpolate directly between the spectrums is an example of "treating the symptom rather than the problem."

The time-varying spectrum of one analysis file may be interpreted as a trajectory through a multidimensional timbral space. This timbral space includes dimensions such as brightness, attack sharpness, noise amount, spectrum density, etc. Then the problem of morphing one sound into another might be best described as finding a trajectory through the multidimensional timbral space of one file to the trajectory of another analysis file's timbral trajectory. It is likely that computing morphing spaces in a higher dimensionality would be computationally prohibitive. Additionally, only a few dimensions of timbre have been codified as particularly salient to timbre identification [Wessel, et al.]

Given the trajectories of two sounds through timbral space, there exists a space of timbral coordinates between the two trajectories. For the purposes of this discussion, this space will be called the morphing space because it represents the area between the two trajectories that the two trajectories have in common. Choosing a point along the morphing surface between the two trajectories is equivalent to selecting a timbre that is similar to the timbre of either source sound based on the position of that point along the morphing space. This relationship between the trajectories and the morphing space might be considered a similarity metric [Polansky] between the two timbres. Ideally, modeling a flexible dynamic system based upon the time-varying analysis spectrum would allow for changes in the dynamic system that would traverse the morphing space towards a destination timbre trajectory.

The Lemur software resynthesizes analysis files with banks of oscillators responding to the peak information contained in the analysis file. An alternative to using the Lemur software for resynthesis of an analysis file is to use a MIDI instrument. Resynthesis with MIDI has the advantage that the MIDI instrument may be configured to resynthesize with waves other than sine tones. The Compile To MIDI operation allows a Lemur analysis file to be transformed into a series of Midi Note-On messages followed with Controller 7 (Volume) and Pitch-Bend information. This operation is ideal for a synthesizer set up to act as a bank of 16 oscillators, playing sine tones (as in a Lemur resynthesis), or more interestingly with other waves such as bells, formsants, or partials. Playing back tracks over MIDI is a coarse approximation
of additive synthesis on a synthesizer that is not equipped for this purpose. A track's starting note-on key is calculated and any timing of the frequency is therefore with Pitch Bend messages to the channel.

The Compile To MIDI operation allows for a few variables to be entered by the user to determine which tracks from the analysis file are chosen for playback. The selection process is crucial to the success of a MIDI resynthesis because only 16 "tracks" may be oscillating at one given time. This increases the responsibility of choosing the "right" track to play. The Compile To MIDI algorithm currently sorts all tracks by track maximum amplitude (highest to lowest). This ensures that the louder tracks will be included in the MIDI resynthesis of the Lemur analysis file.

TWO-FILE MODIFICATIONS

Convolution has been a known filtering technique in signal signal processing for quite some time (Gold and Rader, 1969), and the use of the computer music community has just begun to understand the implications and usefulness of convolution. Traditional convolution involves measuring the entire frequency spectrum (using an FFT or similar technique) of the "impulse" sound and then applying that spectrum to a series of STFT (Short-Time Fourier Transformations) of the source sound to filter it's spectrum by the impulse spectrum. This technique of convolution has been popularized to the audio community by the widespread dissemination of the Macintosh application, Soundtrack by Tom Erbe.

While traditional convolution techniques may alter some spectrum characteristics of one sound to combine with another, the dynamic properties of the impulse sound are lost. For instance, a glissenspiel used as an impulse will be averaged in a traditional convolution, and the exponential decay of the frequency and amplitude envelopes of its harmonics will be lost in the transformation. QuickMQ uses a variant of convolution to achieve a time-varying or dynamic convolution. By applying successive frames of spectrum from the impulse file to the corresponding spectrum frames in the source file, the dynamic properties of the impulse spectrum are preserved in their application to the source spectrum over time.

The Dynamic Convolution transformation "filters" one analysis file by another by multiplying frequency magnitude of source and impulse analysis files (convolution is the time-domain is equivalent to multiplication in the frequency domain (Moore).). Convolution is slightly more tricky on Lemur analysis files than with FFT frequency bins (as used in Soundtrack). FFT frequency bins may easily be multiplied together, as they represent frequency energy around that bin. However, Lemur reduces this data into a series of discrete peaks that do not represent energy in a region, but energy at a single prominence along the discarding the side-bands of frequency energy. QuickMQ must "fill in the holes" with an interpolation scheme when matching source peaks with impulse peaks.

The matching algorithm tries to find two different frequencies in the impulse file for a given peak frequency in the source file. The interpolated magnitude, or inferred magnitude at the corresponding location in the impulse file is calculated using the difference values from the two surrounding peaks from the impulse file as coefficients for the interpolation. The Dynamic Deconvolution transformation works analogously to the Dynamic Convolution transformation, except with an arithmetic divide (÷) rather than an arithmetic multiplication (×). The Dynamic Spectrum Mix transformation works analogously to the Dynamic Convolution transformation, except with an arithmetic addition (+) rather than an arithmetic multiplication (×). The Dynamic Spectrum Subtraction transformation works analogously to the Dynamic Convolution transformation, except with an arithmetic subtract (-) rather than an arithmetic multiplication (×).

MANY-FILE MODIFICATIONS

Translating visual transformations into audio transformations presents a unique challenge. Gerard Holzman'p Popi language allows new images to be computed based on intensity and position information from one or more source images. In short, a corollary from Z = f(x,y) in image space, where Z=pixel intensity, and [x,y] are the coordinates of the pixel must be described in order to accommodate the Popi language within the domain of time-based frequency frames. This mapping can be described as: Z = amplitude of a frequency component at [x,y] in frequency y at time x. Underlying this mapping is a recognition of two strong cognitive factors: 1) The QuickMQ software acknowledges that one frame of audio does not lead to a timbre or source identification for the listener, as one frame of an image can lead to identification of multiple sources by the viewer. However, there exists a subset of the audio frames that lead to multiple source identification as in the case of the single frame of an image. 2) A single pixel Z(x,y) is only meaningful to the identification of the source in the i C M C P R O C E E D I N G S 1 9 9 5 313
context of other pixels, just as QuickMQL treats a frequency component's intensity $A = [f(t)]$ as meaningful only within the context of other frequencies in some frame in a subset of the frames over time. The grammar of the Popi compiler contained in QuickMQL understands commands of the form

$$expression = expression$$

where expression describes a peak vector at a specified time. Values returned by expressions are floating-point vectors of either four or one dimensions. A specific vector may be obtained by using the vecf(f, p, m, t) construction ($f = frequency, p = interpolated phase/frequency, m = magnitude, t = track, 9999 value indicates to "leave alone"). The grammar of the Popi language's expressions may be understood more fully by referring to Gerad Holzmann Popi language (1988). As an example, to shift the spectrum of a sound up by 200Hz, the corresponding Holzmann expression would be:

$$new(x,f) = old(x,f) + vecf(200, 200, 9999, 9999)$$

The above expression reads, for every time and peak in the new analysis file, calculate its value ($=$) by going to the corresponding time and peak of the opened analysis file and add 200 to the frequency of the peak and then add 200 to the interpolated phase/frequency while leaving magnitudes and tracks untouched."

The operators old and new refer to the output or input file. The input file is hard-coded to be the first Lemur analysis file opened by QuickMQL. Future versions of QuickMQL will include a more robust expression compiler and a larger library of supported mathematical and statistical functions.

CONCLUSIONS

The QuickMQL software is capable of producing exciting new sounds out of analysis files. The varied sounds produced by QuickMQL range may range from subtle variation in timbre to dramatic transformations. The author has used QuickMQL to create numerous musical and useful timbres for composition.

Finally, there are specific categories of analyzed sound sources that work best with different QuickMQL operations. Specifically, QuickMQL frame-based and track-based operations bring the most satisfactory results when used with harmonic, steady-state analysis files that have long track formations. This can be maximized by using the hysteresis and threshold parameters in Lemur when analyzing a sound. If a complex analysis file (with several tones, sources, and timbral changes) is used, then frame-based modifications work better than track-based modifications.

The author would like to acknowledge the gracious assistance of Jon Appleton, Charles Dodge, Kelly Fitz, and Max Mathews with the preparation of the software and manuscript for this project. The MQ analysis technique is patented property of the Massachusetts Institute of Technology (MIT). QuickMQL is available via anonymous ftp to osys.dartmouth.edu in the public directory.

REFERENCES


Ebersman, David and David Wessel. "Perception of Timbral Analogies." 13/78: IRCAM.


Fitz, Kelly; Brian Holloway and Bill Walker. "Lemur 3.0.2" software and manual for the Macintosh computer. CERL Sound Group, University of Illinois, 1994.


Wessel, David L. "Low Dimensional Control of Musical Timbre." 12/78: IRCAM.