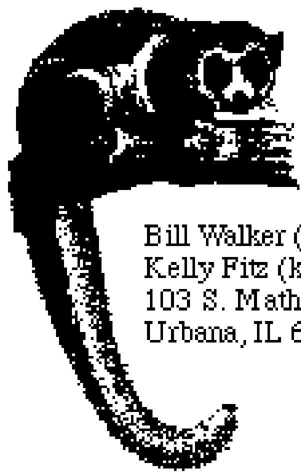
Brought to you by the CERL Sound Group at the University of Illinois



Lemur

Bill Walker (walker@cs.uiuc.edu)
Kelly Fitz (k-fitz@uiuc.edu)
103 S. Mathews
Urbana, IL 61801-2977

Acknowledgements

Lemur is based on `mqan`, a tool developed by Robert Maher and James Beauchamp at the University of Illinois.

Macintosh gurus—Kurt Hebel, John Brewer, Dan Walkowski

Sinusoidal modeling guru—Lippold Haken

Overview

Sinusoidal modeling

Lemur is based on the technique of sinusoidal modeling. The premise of sinusoidal modeling is that any periodic signal can be represented as a summation of sine waves of various amplitudes, frequencies, and starting phases. In the past, most implementations of sinusoidal modeling of non-periodic signals have performed a series of Short Time Fourier Transforms (STFT's) to determine the amplitudes, frequencies, and phases of the sinusoidal components for small segments of the signal. These amplitudes, frequencies, and phases are then used to synthesize a small segment of the signal, and the small segments are strung together to construct the complete, synthesized signal. This straightforward sinusoidal implementation has several problems. The STFT does not report exact frequencies for peaks in the frequency spectrum of the signal. An N-point STFT only reports frequency components at multiples of samplingFrequency / N, limiting the usefulness of this model for frequency scale modification of a signal. Phase uncertainty and discontinuities can create audible artifacts when the synthesis segments are strung together, limiting the usefulness of this model for time scale modification of a signal. (A more detailed description of the STFT may be found in signal processing texts.) Pitch tracking methods solve some of these problems, but their use is restricted to the class of monophonic, strongly harmonic signals. (Grey, Haken)

McAulay-Quatieri technique

In a 1985 technical report from M. I. T. Lincoln Labs, McAulay and Quatieri proposed a sinusoidal analysis technique for speech processing. The premise of the MQ technique is that a sound can be represented by a collection of sinusoidal components (called tracks), each with time-varying amplitude and frequency. To construct these tracks, STFT's are performed on a signal at regular intervals, called frames. Amplitude peaks in the resulting frequency spectra are identified, and parabolic interpolation is used to obtain a close approximation of the exact spectral peak frequencies. These peaks are the most prominent frequencies in the sound at that instant. The peaks in adjacent frames are compared and peaks of similar frequencies are matched. A continuous chain of these matched peaks is a track. To ensure smooth tracks, the MQ analysis minimizes the difference between the frequencies of the peaks being matched. A peak that is not matched represents the birth or death of a track. MQ synthesis uses cubic phase interpolation to reduce phase uncertainty and eliminate phase discontinuities. (Quatieri and McAulay)

Lemur

Lemur uses the MQ technique of creating sinusoidal tracks by linking interpolated peaks in the short time spectra of a signal, and performing cubic phase interpolation (also called "phase unwrapping") during synthesis. Lemur provides some extensions to the basic McAulay-Quatieri technique.

Frequency Bins

The original MQ technique attempted to model psychoacoustic masking effects by suggesting that an amplitude threshold for peak detection should be based on the loudest peak in each frame. In other words, when the sound is loud, only loud peaks need to be represented, since quieter ones will be masked. When the sound is quiet, the quieter peaks are much more important. Unfortunately, this global threshold ignores the importance of frequency in masking effects. For example, a high frequency rarely masks a low one. (Haken, Sciarabba) Lemur provides a refinement of the original MQ amplitude threshold by breaking the frequency domain into logarithmically-sized bins. The loudest peak in each bin is determined, and an amplitude threshold for each bin is based on its loudest peak. This allows quiet peaks to be ignored in a bin containing loud peaks, while detecting quiet peaks in a bin without loud peaks. This is not an accurate model of the effect of frequency in masking, but is an approximation which presents a significant improvement over the original model. A psychoacoustically accurate model is computationally prohibitive, and may not yield perceptibly more accurate syntheses.

Hysteresis

In examining the results of an MQ analysis, one often observes a track which dies out and another track which is born a few frames later at roughly the same frequency. These are best understood as two portions of the same track. Earlier attempts (Serra, 1989) to facilitate this representation allowed tracks to lie dormant for a specified number of frames before dying out. A dormant track had zero amplitude, but participated in peak matching. The dormancy representation gave a more intuitive and visually-pleasing graph of the analysis, but did nothing to reduce the audible effects of low amplitude tracks repeatedly dying and being reborn (this effect has been affectionately called the “doodley-doo” effect). Lemur reduces this effect by allowing the specification of a track amplitude hysteresis. This is the amount by which a track may dip below the amplitude threshold while still participating in synthesis. Hysteresis differs from dormancy in that the tracks in the hysteresis range are synthesized at the amplitudes reported from the frequency spectra, rather than at zero amplitude.

Parameters

Lemur has many parameters which may be adjusted to produce a good analysis (i.e. an analysis which yields good syntheses). Lemur initializes these parameters at startup to values which seem to work reasonably well under a variety of conditions. The optimality of the parameter settings depends heavily on the samples file being analyzed, the available storage, and the patience of the user . The user is therefore encouraged to liberally adjust the parameters from their default values, and to try several different sets of parameters to find the best settings for a particular analysis. An explanation of Lemur's parameter set is included below. At any time, the parameters may be reset to their initial values using the RESET PARAMETERS menu option.

Analysis

During analysis, Lemur analyzes an AIFF samples file (described in Inside Macintosh, volume VI) and creates a Lemur analysis file. While the analysis is in progress, the Lemur Graph window displays the tracks as they are created. The smaller window gives you an idea of how long you will have to wait.

Spectrum

The Spectrum parameters box allows you to control the parameters of the Fast Fourier Transform (FFT) used to compute the frequency spectrum of the signal.

FFT LENGTH • In order to take advantage of various symmetries, the FFT length (the number of points in the transform) in Lemur is always a power of two. FFT length is always a tradeoff: A longer FFT gives more frequency accuracy, but averages over a longer period of time, which means poorer representation of transients. Three choices of FFT length are offered (512, 1024, and 2048 points). Other choices of FFT length give insufficient time or frequency resolution. The default FFT length is 1024 points.

WINDOW LENGTH • The input to each FFT is a windowed set of samples from the input file. A window is used to reduce the spectral artifacts caused by analyzing a segment of samples from a longer signal. The tradeoffs discussed above for FFT length also apply here. The window length must be smaller than the FFT length. The difference between the FFT length and the window length is made up by zero padding (filling out the remainder of the FFT with zeros). This results in spectral interpolation, which makes the peaks easier to find. Best results are obtained by zero padding by at least a factor of two (i.e. the FFT is at least twice as long as the window).The default window length is 512 samples.

KAISER WINDOW PARAMETER • Lemur uses a Kaiser windowing function. The Kaiser window has a parameter which shapes the window, and further affects the time-frequency resolution tradeoff. The default value of 2.0 seems to work well. Values outside the range 0.5 to 4.0 do not. (A brief description of the Kaiser window and the effects of the Kaiser parameter is given in Maher (1989)).

HOP SIZE • Between frames, Lemur “hops” over by a specified number of samples. The hop size is generally smaller than the FFT length, so that each sample is involved in several FFT’s. This is especially important when you intend to time stretch during synthesis. Considerable overlap yields the best results, but also longer analysis times and larger analysis files. The default hop size is 64 samples.

Peak Selection

The Peak Selection dialog box controls various aspects of peak selection and track formation.

NUMBER OF FREQUENCY BINS • Lemur divides the frequency spectrum generated by the FFT into a number of frequency bins, and performs peak detection on each bin separately. The bins are logarithmically-sized, with the smallest bins containing the lowest frequency peaks. For an 1024 point FFT, the following bin arrangements are possible (recall that only the first 512 points will correspond to frequencies below the Nyquist rate):

# of bins	bin sizes
1	512
2	256 + 256
3	128 + 128 + 256
4	64 + 64 + 128 + 256
5	32 + 32 + 64 + 128 + 256
6	16 + 16 + 32 + 64 + 128 + 256
7	8 + 8 + 16 + 32 +64 + 128 + 256
8	4 + 4 + 8 + 16 + 32 + 64 + 128 + 256

Lemur's peak detection algorithm requires that each bin contains at least four points in the frequency spectrum. A longer FFT gives greater frequency resolution and allows more frequency bins to be used. Conversely, a shorter FFT cannot use as many frequency bins, because Lemur will not be able to detect peaks in the smallest bins. In general, best results are obtained by using as many frequency bins as possible for the desired FFT length (e.g. eight bins for an FFT of length 1024). The default (corresponding to an FFT length of 1024) is 8 frequency bins.

MAXIMUM NUMBER OF SIMULTANEOUS TRACKS • Lemur can impose an absolute hard limit on the number of tracks which can sound simultaneously. Lemur will limit the number of births allowed in any frame to the difference between the maximum number of simultaneous tracks and the number of tracks continuing from the previous frame to the current frame. When there are more potential births than can be allowed, the peaks corresponding to those potential births are sorted according to their threshold clearance, and the peaks with the greatest clearance are chosen to be born until the maximum number of tracks is reached. Some synthesis programs have a fixed number of oscillators at their disposal, and this feature may be used to create analysis files to be synthesized by such programs. But such a limit almost always results in a lower quality synthesis. Since Lemur can synthesize any number of simultaneous tracks, the limit should generally be set very high, so that it will have no effect (the default value of 800 is sufficiently high).

BIRTH THRESHOLD • In addition to the time-varying threshold for peak detection presented in the original MQ algorithm, Lemur imposes an absolute birth threshold. This constant threshold is meant to represent the noise floor of the signal. The quietest allowable birth is always determined by the maximum of the two thresholds. Tracks may have amplitude lower than this threshold, due to hysteresis, but a track will never be born at an amplitude lower than this threshold. The default threshold is -25 dB.

RANGE • Lemur locates the loudest peak in each frequency bin and subtracts the Range from its amplitude to determine the peak detection threshold. Reducing the range generally means an overall reduction in the number of peaks, but can have a detrimental effect on the quality of the synthesized sound. The default range is 15 dB.

HYSTERESIS • This is the amount by which a peak corresponding to a track continuation may fall below the time-varying threshold computed from the Birth Threshold and Range. Peaks are detected in this range, but if they fail to match to an existing track, they are discarded. The default hysteresis is 0 dB (no hysteresis).

CAPTURE RANGE • Lemur imposes a neighborhood for peak matching which varies with frequency. The capture range is expressed as a percentage of the matching frequency (i.e. when matching to a 500 Hz peak, with a capture range of 1.0%, only peaks with frequencies between 495 Hz and 505 Hz will be considered). The capture range may be understood as the amount by which a track will be allowed to vary in frequency over the duration of one frame. The default capture range of 2.0 percent seems to work well for most signals.

Synthesis

During Synthesis, Lemur examines an MQ file and produces a new AIFF samples file. While the synthesis is running, the Lemur Graph displays the tracks as they are read from the MQ file.

Scaling and Shifting

One of the most important features of sinusoidal modeling techniques is the ability to perform various transformations which are difficult in the time domain. Lemur provides three transformations during synthesis.

TIME SCALING is performed by adjusting the temporal spacing between frames. Unlike playing your 33 RPM records at 45 RPM, Time Scaling changes the duration of your sound without changing its pitch. If lengthening your sound produces strange frequency artifacts, try re-analyzing with a smaller hop size.

FREQUENCY SCALING multiplies every frequency in the MQ file by a given value before synthesizing. This preserves frequency ratios, but does not model the fixed formants of speech. As a result, frequency scaled speech will not sound like the original speaker talking at a higher pitch. Note that frequency scaling does not change the duration of your sound. If you scale up your frequencies too far, some of them will fall above the Nyquist rate and may cause aliasing.

FREQUENCY SHIFTING adds a given value to every frequency in the MQ file before synthesizing. This does not preserve frequency ratios, but (for small shifting values) tends to preserve fixed formant structures.

Control Files

The three transformations discussed above can be controlled in two different ways—by specifying constant values or by specifying a control file. When a control file is specified, Lemur reads a new value from the control file for each frame. Note that the control file must have the same duration as the original sample file. Thus, the control file provides time-varying control over shifting and scaling. You can construct these control files (which are ordinary AIFF samples files) using whatever tools you have, or with Lemur's simple breakpoint editor, which can generate files with step functions or linear segments. A limit on the amount of scaling or shifting affected by a control file is set using the CONTROL DEFAULTS option. The limits may be set independently. These limits only effect scaling and shifting from control files, and impose no limit on the specification of constant values for scaling or shifting.

Control files are interpreted in several ways, depending on what they are used to control. When a control file is used for time scaling, the samples in the control file are normalized to values between +1 and -1. The time scale is the Maximum Time Scale From File (specified in CONTROL DEFAULTS) raised to this power. Thus, if the Maximum Time Scale is set to ten (10) you can expand or contract by a factor of ten. A control file sample value of zero has no effect. The same algorithm is used for frequency scaling, so you can multiply or divide your frequencies by a factor of ten, when the Maximum Frequency Scale is set to ten. Again, a sample value of zero has no effect. When a control file is used for frequency shifting, the samples in the control file are normalized to plus or minus the Maximum Frequency Shift. As usual, a sample value of zero has no effect. The default maximum scales values are 10.0, and the default maximum frequency shift is 11025 Hz.

The control files facility should prove useful to composers seeking novel effects. Time-varying time scaling is also useful for research in speech processing. Constant time scaling does not model what happens when people speak slowly, since the consonants are stretched along with the vowels. If you construct an algorithm which examines the Lemur file and decides which parts are vowels and which parts are consonants, you can automatically construct a control file which will stretch the vowels while leaving the consonants untouched. Similar ideas might be used to stretch the sustain of a violin tone while leaving the attack transient untouched.

Control files may be generated using the MAKE CONTROL FILE menu option. You will be prompted for a file (Lemur or AIFF format) to control. The length of the control file is determined from this file selection. You may then specify a series of up to ten time and amplitude points for the control file. The control file generated by Lemur is an ordinary AIFF file, and any single-channel AIFF file of the appropriate duration may be used as a control file.

Synthesis File Parameters

By default, Lemur will create a synthesis file using 16 bit samples, at the sample rate of the AIFF file from which the Lemur file was produced. Lemur can also create 8 bit samples files, if desired, for use with other Macintosh applications (the Macintosh has 8 bit D/A converters) using the SAMPLE SIZE option (however, the synthesis file will still be an AIFF file, not an SND resource). The sample rate of the synthesis file may be set independently of the original file using the SAMPLE RATE option.

MQ file format

The MQ file produced by a Lemur analysis begins with a header which contains the following information:

2 bytes	signed integer	file format number
2 bytes	signed integer	header length
12 bytes	floating point	input scale factor
12 bytes	floating point	analysis threshold
12 bytes	floating point	analysis range
2 bytes	signed integer	number of frequency bins
2 bytes	signed integer	FFT length
2 bytes	signed integer	window length
2 bytes	signed integer	analysis hop size
4 bytes	unsigned integer	original number of samples
12 bytes	floating point	analysis sample rate
12 bytes	floating point	capture range
12 bytes	floating point	Kaiser window parameter
2 bytes	signed integer	maximum number of tracks
12 bytes	floating point	hysteresis

Following this header, the file contains successive frames in chronological order. At the beginning of each frame is a two byte integer which specifies how many peaks are in that frame. Empty frames are allowed. The peaks are presented in order from lowest frequency to highest. Each peak has the following format:

12 bytes	floating point	magnitude
12 bytes	floating point	frequency
12 bytes	floating point	phase
2 bytes	integer	next peak index

The next peak index is used to connect a peak with the next peak in its track by giving the ordinality of a peak in the next frame. For example, if a given peak has the value 5 for its next peak index, then that peak should be connected with the fifth peak in the next frame.

The file format number for Lemur 1.1 is 2155. This number may change for future versions of Lemur. Lemur analysis files are of type MQAN and their creator is LEMR.

File Information

Information about Lemur analysis files and AIFF samples files may be obtained using the GET FILE INFO menu option. File size and creation date are given for both types of files. Analysis parameters are given for Lemur analysis files, along with information about the original samples file. Sample size and rate and file duration are given for AIFF samples files.

Lemur automatically updates analysis files which use file formats from earlier Lemur releases. Whenever SYNTHESIS or GET FILE INFO is attempted on an old analysis file, Lemur will offer to update the file. Old files may be updated without using SYNTHESIS or GET FILE INFO by using the UPDATE LEMUR FILE menu option.

Lemur Display

While an analysis or synthesis is in progress, the Lemur Graph window displays the tracks from the analysis as they are generated (during analysis) or read from the disk (during synthesis). Note that during synthesis, the tracks are displayed as they were originally computed, not as they will be synthesized. Therefore, scaling and shifting are not reflected in the Lemur Graph.

The Display menu allows the user to turn off the track display, if desired, or to display a spectrogram of an analysis in progress. DISPLAY SPECTROGRAM only works on color or gray-scale monitors. If the monitor is in Black and White mode, DISPLAY SPECTROGRAM will be disabled. DISPLAY SPECTROGRAM only works during an analysis, because, of course, during a synthesis, the track display is itself an accurate spectrogram of the synthesis file. Lemur may run slightly faster when both displays are turned off. Lemur will run very much slower when the spectrogram display is turned on.

The Lemur Graph window may be closed at any time by clicking in its close box, as with any other Macintosh window. It may be recalled by selecting either of the options in the Display menu.

Known Problems

Lemur may behave strangely if the program is run from a volume mounted with AppleShare or if you attempt to analyze a file on a remote volume. Moving the Lemur Graph window between monitors is not recommended.

References

Robert Crawford Maher, *An Approach for the separation of voices in composite musical signals.* Ph. D. dissertation, 1989, Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign.

John Grey, *An Exploration of Musical Timbre.* Department of Music Report No. STAN-M-2, Ph.D. dissertation, 1975, Department of Physiology, Stanford University.

Lippold Haken, *Real-time Fourier Synthesis of Ensembles with Timbral Interpolation.* Ph. D. dissertation, 1989, Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign.

T. F. Quatieri and R. J. McAulay, *Speech Analysis/Synthesis Based on a Sinusoidal Representation.* Technical Report 693, Lincoln Laboratory, M. I. T.

John Sciarabba, *Psychoacoustics in sound* *synthesis*. M. S. thesis, 1991, Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign (also available through the CERL Sound Group).

Xavier Serra, *A system for sound analysis/transformation/synthesis based on a deterministic plus stochastic decomposition.* Department of Music Report No. STAN-M-58, Ph.D. dissertation, 1989, Center for Computer Music Research in Music and Acoustics, Stanford University.