

Sonogram

Acoustical Frequency Analysis Tool

Christoph Lauer

clauer@dfki.de

March 16, 2002

Abstract

Sonogram transformates time-domain based audio information (samples) into the frequency domain using the transformation methods of "Fast Fourier Transformation" (FFT), "Linear Prediction Coefficients" (LPC), "Cepstral Analysis" and the "Wavelet transformation". It extracts samples from multimedia files and daws them into the main window which shows the frequency information in a two-dimensional form (frequency-time area). The most common audio and video file formats are supported. The two-dimensional plots can be shown in an interactive three-dimensional representation. Many signal analysis parameters can be configured and the representation can be configured in different forms. The program is written in Java2 therefore it runs on any operating system where Java2, is supported. Calculated pictures can be stored as "Scalable Vector Graphics" (SVG) and bitmaps. It has a build-in documentation which describe most of software components. The program can be free used for non comercial and research purposes.

Contents

1	Introduction	2
1.1	Background	2
1.2	How to use this Document	2
1.3	System Requirements and the Java Installation	2
1.4	Installation and Starting	2
2	Digital Signal Processing	3
2.1	What is a Sonogram ?	3
2.2	Digital Audio Processing	3
2.3	Signal Analysis and the Fourier Transformation	4
2.4	Windowing the Signal	4
2.4.1	Aliasing in Spectrum Analysis	4
2.4.2	Window Function	4
2.4.3	Choice of Window Function	5
2.4.4	Leakage in Spectrum Analysis	5
2.4.5	Overlapping	5
2.5	Linear Prediction Method (LPC)	6
2.6	Frequency-domain Pitch Detectors	6
2.7	Cepstral Analysis	6
2.8	Analysis by Wavelets	7
3	Concept	8
3.1	Opening, Calculating and Drawing	8
3.2	Supported File Formats	8
4	User Interface	8
4.1	Main Window	8
4.2	Adjustment Dialog	9
4.3	Toolbar and Menu bar	10
4.4	Analyse Windows	11
4.4.1	Waveform Display	11
4.4.2	Fourier Display	11
4.4.3	LPC Display	11
4.4.4	Wavelet Transformation	11
4.4.5	Cepstrum Display	12
4.5	Information Dialog	12
4.6	3D Surface Plot	12
4.7	File Open Dialog	12
4.8	Online-Help	12
4.9	Keyboard Shortcuts	13

5 Working with Sonogram	13
5.1 Opening a File	13
5.2 Configuring the Sonogram with the Adjustment Dialog	14
5.3 Analysing Windows	14
5.4 3D Surface Plot	14
5.5 Working with SVG	14
6 Future Work	14
A Musical Frequencies	15
B About this Document	15

1 Introduction

1.1 Background

Sonogram arose from a spectral analysis module for Michael Kipp’s annotation program “Anvil“ (www.dfki.de/~kipp/anvil), at the “German Research Center for Artificial Intelligence” (DFKI) in Saarbrücken. The first task of this module was to transform audio files in a frequency spectrum. Bit for bit more features were added to the program, resulting Sonogram in the current version. It analyses audio information in the frequency-domain with the most common methods used in phonetics and linguistics, and presents them graphically. Since 1965 Cooley and J.W. Tukey reduce the “Fast Fourier Transformation“ (see chapter 2.3) algorithm, computer provided frequency analysis has made great advances. Sonogram’s aim is to be as configurable as possible for the different usecases of spectral analysis in speech and sound analysis.

1.2 How to use this Document

This document tries to be as complete as possible and describes the use of Sonogram. It also gives an introduction into the digital signal analysis. The “Digital Signal Processing” chapter shows the more theoretical background of the program and makes clear the view from the point of the signal analysis. The “User Interface” section describes all functions in a more detailed form. A

more practical point of view gives the “Working with Sonogram” chapter.

1.3 System Requirements and the Java Installation

Sonogram needs Java2 in the version 1.3 or higher, Java Media Framework (JMF) and Java3D. For Solaris and Windows systems JMF can be downloaded from Sun Microsystems (www.java.sun.com). The Java Media Framework for Linux platforms is available from Blackdown (www.blackdown.org). The installation for Linux is done by copying the jar files into the right folder how described in the readme file which comes with the installation package. Apple OS-X users should use the cross platform version of JMF and copy the jar files in the /Library/Java folder or add the jar files to her classpath. The Java 3D installation is optional. MS-Windows have the choice between the DirectX and the OpenGL version at Sun’s website. Windows NT 4 users should use the OpenGL version, the rest can download the DirectX version. To install Java3D to the UNIX derivations simply unpack the tar archive into the Java installation folder and set the classpath how described in the readme file. At this time a Java3D version for Apple OS’es is not available therefore the 3D surfaceplot function is not available for the Macintosh operating systems.. It was successfully tested on Linux, Solaris 7, Mac OS-X and all Windows platforms. The processor speed should be greater than 100MHz and the system should have at least 64MB of physical RAM.

1.4 Installation and Starting

Sonogram can be downloaded from DFKI’s NITE Project page www.dfki.de/nite, from www.dfki.de/~clauer/sonogram or from the authors home page www.christoph-lauer.de. The **Sonogram.zip** file includes the **Sonogram.jar** class archive, start scripts for UNIX, Mac OS-X and Windows and various icons. The unpacked files should be placed into a new directory. On Windows systems, Sonogram is started with the start script **Sonogram.bat**. Unix and Linux users change the file attributes with **chmod u+x**

`Sonogram.sh` and start it with `./Sonogram.sh`. Macintosh OS-X users should open the program simply by clicking the jar archive. Sonogram can also be started manually from a console with `java -classpath Sonogram.jar:. Sonogram`. The program does start without installed Java3D, without JMF the start process will end with a message. Without the Java3D installation the 3D surface plot function is disabled.

2 Digital Signal Processing

This chapter give a short introduction of digital the audio-processing, the signal analysis and the methods of windowing.

2.1 What is a Sonogram ?

A *sonogram*, *sonograph* or *spectrogram* is a well-known spectrum display technique in speech processing and sound analyse, having been used for decades to analyse sounds and voices. A sonogram shows an overview of the spectrum of several seconds of sound. This enables the viewer to see general features such as the onset of notes or phonemes, formant peaks, and major transitions. A trained viewer can read the phonemes in the sonogram. The sonogram representation has also been employed as an interface for spectrum editing and music analysis. The original sonogram was Backhaus's(1932) system [Roa96], described earlier in the background section on spectrum analysis. In the 1950s the Kay Sonograph [Roa96] was standard device for making sonograms. It consisted of a number of narrow bandpass analog filters and a recording system that printed dark bars on a roll of paper. The bars get thicker in proportion to the energy output from each filter. Today sonograms are implemented with the "Windows Fourier Transformation".

2.2 Digital Audio Processing

Digital audio data is commonly stored as time-domain based discrete information. Discretisation of the signal is done in the time and the amplitude dimension. In the time dimension discretisation

(also named *quantization*) depended on the sampling rate, which describes how many sampling points are computed for second. The *Nyquist theorem* defines the maximal frequency that can be rebuild from a digital stored signal. This is the half sampling frequency called the (*Nyquist Frequency*). A CD player which read the samples stored with 44100 samples per second play the signal with maimal $22050 = 44100/2$ Hz. The other quantsitation is take on the amplitude. The values of the samples from the signal cannot take any conceivable value. This is because digital numbers can only be represented within a certain range and with a certain accuracy, which varies with the number of bytes used to store the amplitude. The implications of this are an important factor in digital audio quality. Samples are usually represented as integers. If the input signal has a voltage corresponding to a value between 53 and 54, for example, then the analog digital converter might round if off and assign a value of 53. In general, for each sample taken, the value of the sample usually differs from slightly from the value of the original signal. This problem in digital signals is known as *quantization error* or *quantization noise*. The amplitude quantization error is dependent on two factors: the digital signal itself, and the accuracy the signal is represented in digital form. If the quantization is very fine, for example 24 bit which divides the amplitude into 2 up 24 parts, the quantization error is very small. In test with humans psycho acoustic scientists have found the maximal audible amplitude difference between two signals, and calculate that 12 bit amplitude quantization is enough for this span. To sum up quantization is done in two typs:

- Amplitude quantization, which divides the amplitude in discrete parts and make quantization errors of the amplitude in comparison with the original signal.
- Time quantization, which is presented by sampling rate. They give the stored amplitude numbers per second.

2.3 Signal Analysis and the Fourier Transformation

Classical signal analysis goes back to Jean Baptiste-Joseph Fourier (1768-1830). He described in his work about “Analytical Theory of Heat” [Lau00] that every symbolic function can be displayed as infinite summation of sinus-functions. He developed the “Fourier Integral” which transformed every symbolic function in a summation of sines and cosines functions. For time-domain based real functions, e.g. sound waves, this summation represent the included harmonic frequencies. For discrete datasets the continuous fourier integral pass over to a discrete summation name fourier Series. 1965 Cooley and J.W. Tukey developed in their work ”An Algorithm For The Machine Calculation Of Complex Fourier Series” [Lau00] an algorithm that is very fast if the number of samples is an power of two)¹. This algorithm, named “Fast Fourier Transformation” (*FFT*), or one of his deviations is used in many technologies, for instance: mobile telephone, MP3, geologic research, audio analyse devices... The FFT transformation for a complete audio signal, e.g. 180 seconds with 44100 samples per second, generates a frequency vector with $44100/2 * 180 = 3969000$ frequency points ²in the scale from 0...22050Hz. The frequency resolution for this signal is $22050Hz/3969000 = 000555Hz$. This is a very very good resolution, but at which time the frequency in the signal appears is absolutely not clear. Although the complete information of the signal is stored in this frequency vector it is not possible to make any predication at which time every frequency has his maxima or minima and so on. 1946 Denis Gabor developed the “Windowed Fourier Transformation” (*WFT*) also named Gabor Transformation to elude this disadvantage.

¹see [Bri95] for detailed information about FFT.

²with the idealism for physical real signals without any imagine allotment. Normally the FFT computes complex vectors but for real vectors the outcoming spectrum is mirrored at the middle, so the negative frequencies are not relevant. In this physical case the spectrum is the absolute amount spectrum. Compare [Lau00].

2.4 Windowing the Signal

In signal analysis the term ”windowing” means that a time-domain based signal is splitted into pieces of fixed length. The time resolution is precisely, but still limited to window length. If the 180 second time-domain based signal is splitted into 3000 peaces with $44100 * 180/3000 = 2646$ samples per window the time resolution is $180/3000 = 0.06seconds = 60milliseconds$. The FFT algorithm computes $2646/2 = 1323$ (comp. *Nyquist*) frequency points and consequently an frequency resolution from $22050Hz/1323 = 16.66Hz$. Consequently, when time resolution is going up frequency resolution is going down and vice versa. This fact is the acoustic uncertainty relation, which means that not both units can be defined exactly. The acoustic quant is consequently: $frequencyresolution * timeresolution = constant$.

2.4.1 Aliasing in Spectrum Analysis

Frequency components in the signal over the nyquist frequency can not be strobed correctly.(see Figure 1) They are under-sampled and are mirrored at the head of the spectrum. This is the reason for anti aliasing filters which margin the band width to nyquist frequency before the signal is sampled. Sonogram’s option to open files with 8KHz instead of the original sampling rate has no anti aliasing filter included, so signals with frequencies more than 4KHz generate aliasing effects.

2.4.2 Window Function

The single windows have one disadvantage. They are cut out of the signal, so on the ends the signal is not zero. For frequency transformation this means the signal has frequency peaks on different places. To denoise this window, the window itself is multiplied with any window function like the gaussian, a triangle or the hamming function. The result of the FFT is smoother. All standard (nonrectangular) windows are symetric and have spectra whose shapes resemble a mathematical sinus function, that is $\sin(t)/t$. This shape is char-

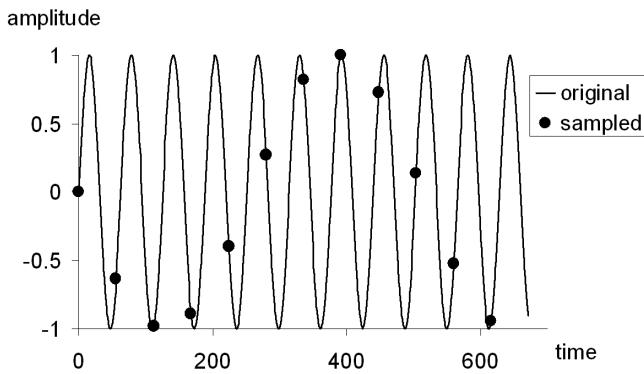


Figure 1: Aliased sampling from a sinus signal.

acterized by a prominent main lobe and a series of side lobes on either side of the main lobe. The main properties of the window rate the width of the main lobe defined as the number of frequencies bins it spans- and the highest side lobe level, which measures how many decibels down the highest side lobe is from the main lobe. The magnitude of the side lobes is dependent on the shape of the window function and is dependent of the window's duration.

2.4.3 Choice of Window Function

From a practical standpoint, there is no "universal best" window choice for all sounds and all analysis purposes. For many musical applications involving sound transformation, a smooth bell shaped curve such as Kaiser or Gaussian window works reasonably well. The advanced user of the phase vocoder must decide which aspect of analysis is most important in a specific application. For example, when it is known that the input signal is composed of a distinct number of sine components, then a window with a narrow main lobe is recommended. For noisy signals, where the energy is not centered at individual frequencies, a window with a broad main lobe works best. Since the choice of window affects the plot, spectrum displays for scientific purposes should indicate the type of window used in the analysis. In the Sonogram adjustment dialog register card "window Function" the lobes damping for the windows functions is dis-

played.

2.4.4 Leakage in Spectrum Analysis

When a single sine wave is fed into an FFT, we would like the energy to be concentrated in the main lobe. Unfortunately the process of the windowing disorts the analysis and causes *leakage*. or "splatter" from the main lobes. Any frequency present in the window signal splatter across all frequencies bins of the output spectrum. This can clearly be seen in the spectrum of a sine wave windowed by a rectangle. Here the window diffuses the single input frequency component into energy extending across the bandwidth of the analyser.

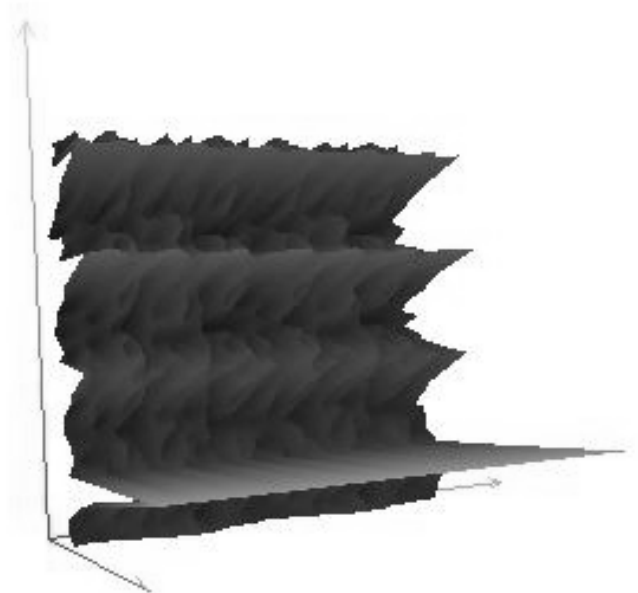


Figure 2: Leakage in spectrum generated with no window function from a 500Hz sinus wave.

2.4.5 Overlapping

Overlapping in signal analysis means that single windows are not sequential one after an other. If overlapping is zero every window follow another with step width from one window to the next. To get more windows for each time span step width is smaller than one window length. The factor of step with to window length is named overlapping. Overlapping can not break the acoustic un-

certainty relation, it brings only more transformations for each time span.

2.5 Linear Prediction Method (LPC)

Linear prediction derives its obscure name from the fact that in the spectrum analysis part of the system, output samples are “predicted” by linear combination of filter parameters (*coefficients*) and previous samples. A *prediction algorithm* tries to find samples at positions outside a region where one already has samples. That is, any extrapolation of a set of samples is prediction. Inherent in prediction is the possibility of being wrong; thus prediction algorithms always include an error estimation. A simple predictor simply continues the slope of differences between the last sample and the sample before it. This type of predictor can be made sophisticated by taking more samples into account. It can also take into account the error or distance between the sample it predicts and the actual value of the signal, if this is known (and it is known in LPC). Since the predictor is looking at sums and differences of time-delayed samples, it can be viewed as a filter—a filter that decides the waveform it is currently processing. If we take regular snapshots of these filter coefficients over time, invert them, and then drive the resulting filter with a rich, wide-bandwidth sound, we should have a good approximation of the time-varying spectrum of the output signal. Thus a “side effect” of the prediction is to estimate the spectrum of the input signal: this is the important point. But spectrum estimation is only one stage of LPC analysis, the others being applied to pitch, amplitude, and the voice/unvoiced decision. Spectrum of so linear estimated signals seems smoother like his FFT brother. In Sonogram the LPC algorithm³ first generate the linear polynome with a configurable number of samples (previous samples) with them the extrapolated time-doamin signal is generated. The degree of the polynome (number of coefficients) is change the precision of this vector in relation of the original. This extrapolated sample vector with the length of the FFT buffer isthen

³See online manual for a further information and a picture of this LPC algorithm

transformed. The result is the LPC spectrum.

2.6 Frequency-domain Pitch Detectors

Frequency-domain pitch detection methods dissect the input signal into frequencies that make up the overall spectrum. The spectrum shoes the strength of the various frequency components contained in the signal. The goal is to isolate the dominant frequency, or *pitch* out of the spectrum. A typical frequency-domain approach analyzes successive segments of the input signal using the short-time fourier transform (WFT). Frequency-domain pitch detectors seek peaks in the spectrum that correspond to prominent frequencies. After finding the peaks, the pitch detector must decide which frequencies are fundamentals (general perceived as pitches) and which frequencies are merely harmonics or extraneous partials. A quick real-time frequency-domain pitch detector might simply select the strongest frequencies as the pitch. A more sophisticated detector might look for harmonic relationships that imply a fundamental frequency. This fundamental may not be the strongest component, but it may be the most prominent perceived pitch due to the “reinforcement” of multiple harmonics. A problem with WFT based pitch detectors is that the WFT divides the audio bandwidth into a set of equally spaced frequencies *channels* or *bins* where each channel is n HZ apart from its neighbors. Since human pitch perception is basically logarithmic, this means that low pitch may be tracked less accurately than high pitches. For example , ana analyser with frequency resolution of 20 Hz can resolve microtones in the register between 10 and 20KHz, but offers less than a semiton resolution below middle C. Accurate pitch resolution at the low end of the spectrum of analysis channels.

2.7 Cepstral Analysis

A common frequency-domain pitch detection method in speech research is the cepstrum technique, having first been used in analysis of speech. Cepstrum analysis has often been applied in conjunction with the technique of linear predictive coding (LPC). The term “cepstrum” war formed

by reversing the first few letters of “spectrum”. A simple way of describing the cepstrum is to say that it tends to be a reasonable model of many vocal and instrumental sounds whose spectrums can be considered as the sum of an *excitation* (the original vibrational impulses, typically at the pitch of the sound) and the *resonances* (the filtered part of a sound created by the body of an instrument or the vocal tract). Technically, the cepstrum is the inverse fourier transform of the *log-magnitude Fourier spectrum*. This is the absolute value of the log (base 10) of the output of the discrete fourier transform. The result of the cepstrum computation is a time sequence, like the input signal itself. If the input signal has strong fundamental pitch period, this shows up as a peak in the cepstrum. By measuring the time distance from time 0 to the time of the peak, one finds the fundamental period of this pitch. How does cepstrum analysis work for speech? The cepstrum serves to separate two superposed spectra: the glottal pulse (vocal cord) excitation, and the vocal tract resonance. The excitation can be viewed as a sequence of quasi-periodic pulses. The fourier transform of these pulses is a line spectrum where the lines are spaced at harmonics of the original frequency. The process of taking the log magnitude does not affect the general form of this spectrum. The inverse fourier transform yields another quasi-periodic waveform of pulses. By contrast, the spectrum of the response of the vocal tract (acting as filter) is a slowly varying function of frequency. The process of applying the log magnitude and the inverse fourier transform yields a waveform that has significant amplitude only for a few samples, generally less than the fundamental pitch period. It can be shown that the impulse response decays as a function of $1/n$, then its cepstrum decays as $1/(n^2)$. Thus the cepstrum clusters the impulse response into a short burst at the beginning of the cepstrum wave, and it clusters the pitch into a series of peaks at the period of the fundamental frequency. Cepstrum computation has many applications because it tries to sort out the impulse response from the excitation. In other words, the cepstrum tends to deconvolve the two convolved spectra. The log magnitude oper-

ations in the cepstrum procedure tend to cluster these two quasi-separate components of the spectrum. By advanced operations that we will not describe here, either of these features can be filtered out so that cepstrum contains spectrum information associated with either timbre or pitch. Another application of the cepstrum is found in speech analysis/resynthesis. If there is no peak in the cepstrum this is an indication that the sound being analyzed is unvoiced—that is, it is a breathy or consonant sound with no pitch like “f” or “s”, as opposed to a voiced vowel like “ah”.

2.8 Analysis by Wavelets

The *wavelet transform* (WT) was originally developed by scientists at the University of Marseille for applications in physics and acoustics. A wavelet is a signal that forms a sinusoid with a smooth attack and decay. The term “wavelet” and its French equivalent *ondelette* are not new, however, being used in early-twentieth-century physics to describe the packets of energy emitted by atomic processes. From a signal analysis perspective, the WT can be considered as a spherical case of the constant Q filter paradigm. Wavelets inject the notion of a “short time” or “granular” representation into the constant Q filter model. The WT presents and manipulates sounds mapped onto a time-frequency grid or plane. Each rectangle in this grid presents its *uncertainly product*. The center of each grid is the mean time of occurrence and the spectral centroid. Such a grid is also implicit in constant Q methods (used with digital filters), but it is rarely used explicitly. In music analysis with the WT, one sets up the grid according to the goals of the analysis and disorts the grid according to the goals of the resynthesis. In the wavelet theory, every signal can be expressed as a sum of wavelets having a precise starting time, duration, frequency, and initial phase. The prototypical wavelet for music has a Gaussian envelope, but other types of wavelet envelopes can be defined. Thus the wavelet is similar to the gain and to the windowed segments of the short time fourier analysis (comp. WFT). The peculiar aspect of the wavelet is that no matter what frequency it

contains, it always encapsulates a constant number of cycles. This implies that size (duration) of the wavelet window expands or compress according to the frequency being analysed. This stretching and shrinking is referred to as *dilation* in the literature and is usually specified as a factor of 1/frequency. The implication of the dilating window size is that the WT trades frequency resolution for time resolution at high frequencies, and trades time resolution for time for frequency resolution at low frequencies. Thus the WT can simultaneously detect precise onset times signaled by high-frequency transients, and also resolve the low-frequency spectrum well.

3 Concept

3.1 Opening, Calculating and Drawing

Generating a Sonogram happens in three steps. Samples are read from file, frequency transformation gets calculated with adjusted settings and a picture is drawn on screen. Those steps can be started manually from the adjustment dialog. Usually "Fast Fourier Transformation" is the standard transformation algorithm for frequency analysis in speech and musical frequency analysis and is implemented internally. Alternatively, the "Linear Prediction Coefficients Method" (see chapter 2.5) algorithm can be chosen which generates smoother spectrums than the FFT (see chapter 2.3) transformation. The window length for those transformations can be selected manually or the program may search the best length for optimal displaying on the screen. The transformation results are stored in a matrix for repaint drawing events. For faster displaying areas when window comes in foreground or other windows are moved, the double buffering technology in the program stores the picture internally and refresh the needed zones without calling the paint functions. To speed up displaying, double buffering is used to redraw areas of the window when it is exposed: The picture is stored internally and used to refresh needed zones without having to call the paint function.

3.2 Supported File Formats

With the help of the "Java Media Framework" library the program extract the samples from the multimedia files. The Supported file formats are dependent from the version of JMF. But most common audio and video ⁴ files are supported. The supported file formats are dependent from the operating system version of the JMF. The at this moment the supported file formats for the most operating systems are:

- **AUDIO:** Sun Audio (au), Wave Audio (wav), Apple Audio (aif,aiff,aifc).
- **VIDEO**⁵ AVI Video (avi), Macromedia Flash (swf,spl), GSM Video (gsm), MPEG Layer I,II,III (mpeg,mp2,mp3), Apple Quick Time (mov), IBM HotMedia (mvr),

4 User Interface

4.1 Main Window

The biggest region of the main window show the Sonogram, displayed in the selected color. The Frequency scale can be switched between a linear or a logarithm scale. In the under section of the main window the signal is shown in the time-domain, colored in the self color the sonogram itself is shown. This can show either the amplitude or the energy of the time signal, dependent from the selected option in the adjustment dialog. On left side the frequency numbers are plotted besides the grid if it is enabled. The information displayed in the under right corner, shows information in two areas. The upper area shows the peak frequency of time point under the mouse (**P**), the actual frequency under the mouse (**F**), the time position under the mouse (**T**) and the relative amplitude in relating the maximal amplitude of the displayed sonogram (**A**). The under area shows choices made in the options for the program like window function, logarithm amplitude or window overlapping. The graph of the right side illustrates the current

⁴Only audio information track of video files are analyzed.

⁵Video files can be stored with different CoDdec's. Here performed files must not be support implicit! See Sun's JMF supported file formats for details.

spectrum from the time position under the mouse. Under the menu bar the toolbar with his violet colored buttons is placed.

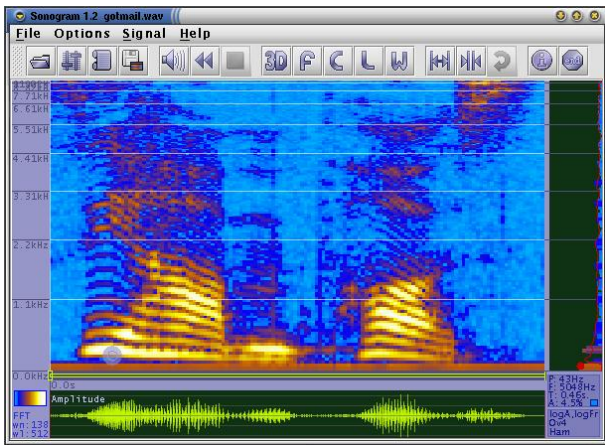


Figure 3: Main Window.

4.2 Adjustment Dialog

The adjustment dialog is partitioned in register cards.

- **General:**

- Spectrum while playing enables the animation of the right side spectrum.
- Monochrome single curve let the right side spectrum shown in violet color.
- Normalize single curve uses the full area of the right side spectrum.
- Smooth out sonogram over frequency axis smoothed the frequency picture horizontal.
- Smooth out sonogram over time axis smoothed the frequency picture vertical.
- Save hot list while ending stores the file history in the `SonogramConfig.xml` file.
- Save screen positions while ending stores the screen positions and window dimensions of all windows in the `SonogramConfig.xml` file.
- Save configuration while ending stores all adjustment informations in the `SonogramConfig.xml` file.

- Loop playing repeat the playing.
- Open last file at start opens automatically the last file when Sonogram started next time.
- Open last file with zoom opened the file with zoom which was selected last time.
- Open with 8KHz sample rate convert all files to 8KHz sample rate. This may be profitable for long files.
- Show grid in Sonogram.
- Show intensity shows the energy instead the amplitude in under partition on main window.
- Search local peak highlighted highest peak in the showed Sonogram.

- **Colors:** Colors for the main window can selected and inverted.
- **Window Functions:** Every window function has a specific attenuation characteristic which is shown in upper area.
- **Window Length:** Automatically windowing search the best length for optimal displaying.
- **Overlapping:** Can be enabled or disabled.
- **Logarithm Amplitude:** Influence coloring.
- **Logarithm Frequency:** Influence the scale of the logarithm frequency graduation.
- **Gain**The The loudness.
- **3D Surface Plot:** Point, line and a solid surface are selectable. Length and width alter the proportions of the wave-mountain Color change the background color and universe switcher enables/disables the background picture. The data reduction slider is in beta phase.
- **FFT:** Window length and scale-width are adjusted here for the Fourier window.

- **LPC:** This settings are explained in the LPC chapter(2.5) of this document and in the online-help where an overview picture is available. This register card changes the settings for the LPC single window and the major settings for the LPC Sonogram if the transformation in selected to LPC ! Is the Sonogram is calculated with the LPC transformation the relevant sliders are highlighted.
- **Cepstrum:** Calculating buffer and logarithm quefrenz (see chapter for Cepstral analysis 2.7).
- **Waveform:** The time span shown in dialog window can adjusted here.
- **Transformation:** The transformation for the Sonogram generating is changed here.
- **Pitch:** If pitch detection is enabled a for can be lay over the Sonogram. The frequency limitation bounded the pitch search into a frequency span from zero to selected frequency. The smooth out function tries to smooth the pitch curve.
- **Wavelet:** The number of otaves and the wavelet for the transformation can be selected here.

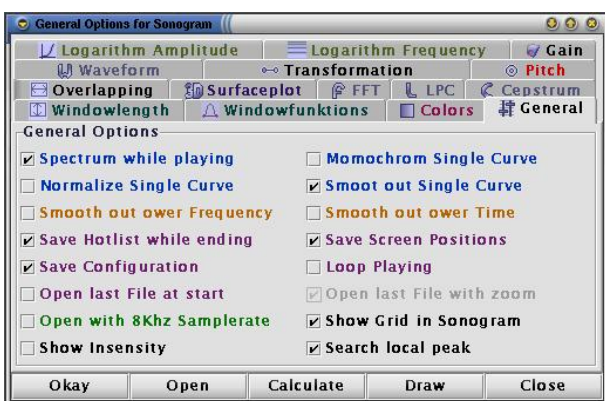


Figure 4: Adjustment Dialog.

4.3 Toolbar and Menu bar

Some items in the menu bar are linked with functions in the adjustment dialog. Only not linked items are listed here.

- **File - open local Media File:** Shows the file-open dialog.
- **File - open local Media File by URL:** Shows the net-file-open dialog. The full referenced URL must entered here.
- **File - open Sonogram from SVG:** If the Sonogram is exported as SVG file, it can be reopened with this dialog. After re-importing the Sonogram the program ask you for reopen original file. If the original file is not available or user cancel reopen not all features of Sonogram program will be available why original signal is not present.
- **File - Save Sonogram as SVG:** The Sonogram can exported to SVG. After file is selected colors and grid can selected. The pure SVG option removes all Sonogram specific information from the SVG file, but re-importing without this information is not possible! Graphic size adjusts the square size of the single pixel.
- **File - Export Sonogram as Bitmap:** Sonogram will be stored as bitmap.
- **File - Print Sonogram:** Prints the Sonogram without single spectrum and information box.
- **File - Exit:** Quits program.
- **Options - Fullscreen:** The Main window is sized to full screen size and toolbar is hidet.
- **Options - Default Settings:** The First dialog ask for reset all settings to default, last dialog ask for resize all windows.
- **Options - Select LookAndFeel:** General outfit for all dialogs can be changed
- **Help - System Information:** Displays detailed informations about the operating system and the java configuration.

- **File - The file history** Sonogram stores all filenames from files they are successfully opened. Non available files have a red line and net files have a green earth in icon.

The Icons in the toolbar correspond with according elements on the menu bar.

4.4 Analyse Windows

The Analyse windows "Wave-Form", "Fourier", "Linear Prediction", "Cepstum" and "Wavelet" are single windows which work separate from the Sonogram picture. All the settings for this windows are made in the adjustment dialog on the adequate register cards. The windows can be resized.

4.4.1 Waveform Display

The waveform window shows the signal as a oscillogram graph in time-domain. Amplitude of the signal is absolute amplitude from the file. The signal walks with mouse over the Sonogram and shows the length selected in adjustment dialog. Is the time window after the mouse at the end of the signal a red information on upper right corner will be displayed.

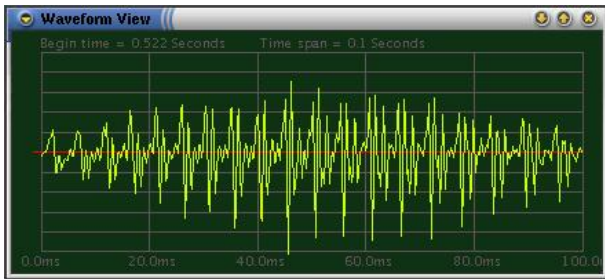


Figure 5: Waveform Window.

4.4.2 Fourier Display

The fourier window shows the fast fourier transformed of the signal under the mouse with the window length selected in the general adjustment dialog. All parameters for this transformation are

adjusted in the analogical register card. The frequency fraction slider scales the x axis of the frequency plot. Is the time window after the mouse at the end of the signal a red information on upper right corner will be displayed.

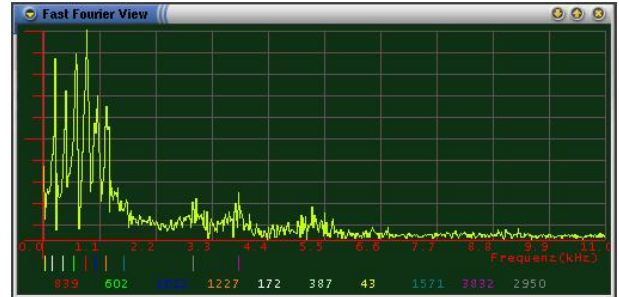


Figure 6: Fourier Window.

4.4.3 LPC Display

The linear prediction coefficients window shows the spectrum of the linear extrapolated signal under th mouse. See in the online-help and in the chapter for LPC for detailed information about this transformation. All parameters can be configured in analogical register card in adjustment dialog. Is the time window after the mouse at the end of the signal a red information on upper right corner will be displayed.

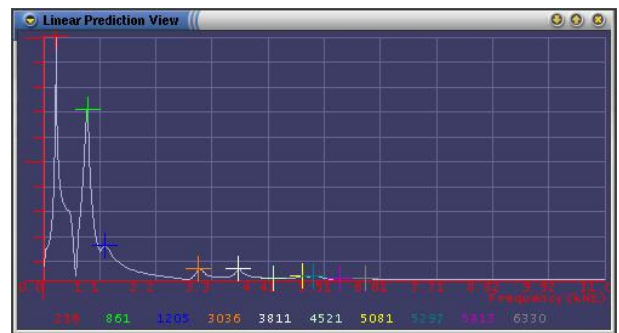


Figure 7: LPC Window.

4.4.4 Wavelet Transformation

The wavelet tranformation window shows the two dimensional time-octave view. The time window

for the transformation begins under the mouse and has the span corresponding to the number of octaves selected in the adjustment dialog. Accuracy the number of samples is the octave power of two.
 $\Rightarrow windowtimespan = \frac{2^{octaves}}{samplerate}$

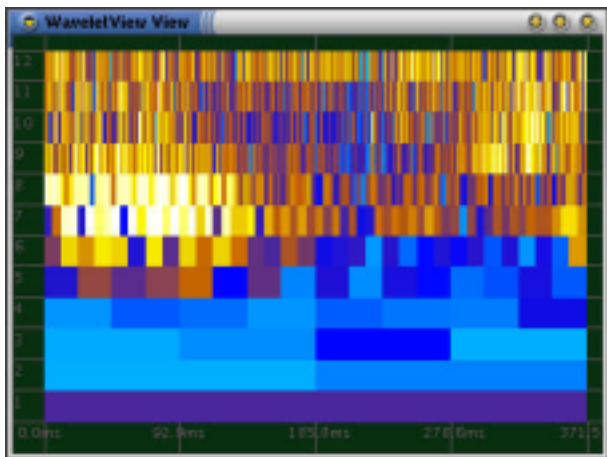


Figure 8: Wavelet Window.

4.4.5 Cepstrum Display

The cepstrum window shows the cepstral transformed signal from the position under the mouse. See chapter cepstral analysis for more details on selected algorithm.

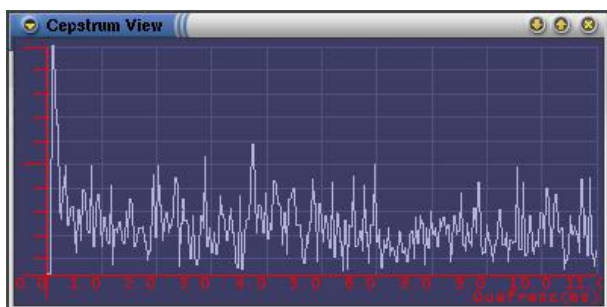


Figure 9: Cepstrum Window.

4.5 Information Dialog

The information dialog (enabled with the corresponding button on the toolbar or the menu item) gives detailed information over the signal, the file

format and all internal selected parameters. All informations are updated immediately when corresponding information has changed.

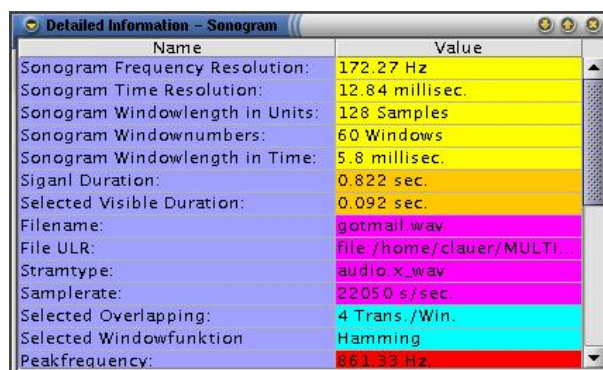


Figure 10: Information Dialog.

4.6 3D Surface Plot

If Java3D is installed on local machine this feature is enabled. All options are made in adjustment dialog. If the Sonogram has more than 20000 points the dialog will ask you if you want to generate this plot. More than 20000 are very slow on video-cards with less than 64MB ram. The generated plot is movable and sizable with mouse. The Right mouse button cycles the surface plot and left button moves them in room. With middle button (both buttons on windows systems) the plot can be zoomed in and out.

4.7 File Open Dialog

File open dialog in Sonogram have enhanced possibility like a standard java file chooser dialog. The included find utility permits searching for files and shows the directory history. Supported file types for JMF 2.1 are listed in the onlinehelp or in the chapter 3.2.

4.8 Online-Help

Info button on toolbar enables online help for Sonogram. He shows additional information for included features. If local computer is connected to internet, blue underlined links can be used.

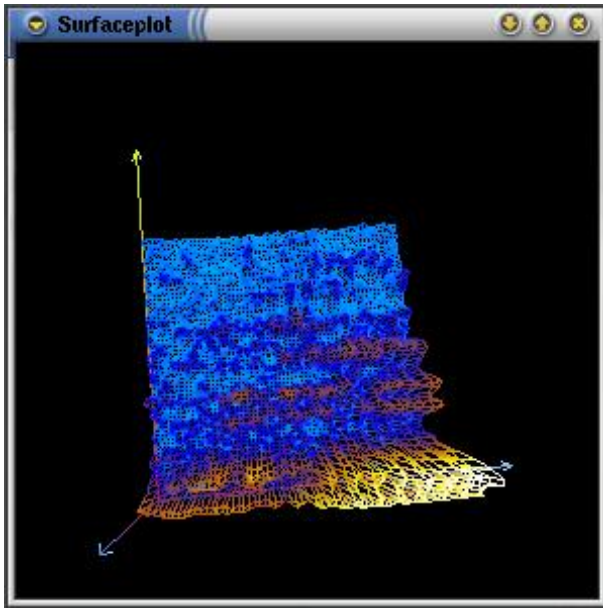


Figure 11: Surface Plot.

4.9 Keyboard Shortcuts

Key	Function
(Strg) + (o)	open file from Disk
(Strg) + (u)	open file from URL
(Strg) + (y)	re-import SVG spectrum file
(Strg) + (v)	save Sonogram as SVG file
(Strg) + (e)	export Sonogram as Bitmap
(Strg) + (n)	open file <i>n</i> from history list
(Strg) + (q)	quit program
(Strg) + (a)	Open adjustment dialog
(Strg) + (f)	select "Look and Feel"
(Strg) + (i)	enables/disables inverse colors
(Strg) + (g)	enable/disable grid in Sonogram
(Strg) + (k)	enable/disable logarithm amplitude
(Strg) + (l)	enable/disable logarithm frequency
(F11)	Fullscreen
(Strg) + (i)	info dialog
(Strg) + (p)	play selected signal
(Strg) + (r)	rewind playing
(Strg) + (s)	stop playing
(Strg) + (3)	show 3D surface plot
(Strg) + (o)	show LPC window
(Strg) + (c)	show cepstrum window
(Strg) + (l)	show waveform window
(Strg) + (H)	show wavelet window
(Strg) + (z)	zoom in marked area
(Strg) + (b)	zoom back to full signal
(Strg) + (v)	zoom back to previous zoom
(F1)	open online-help

5 Working with Sonogram

5.1 Opening a File

Opening the an multimedia file is splitted in two phases. First the samples are read out from the file and second the Sonogram will be calculated with the options adjusted in the configuration dialog. Two times a progress dialog appears in the under edge of the main window. During progress bar advance a mouse click on it will stop the complete opening procedure immediately. The so generated Sonogram occurs on the main window. Mouse movement over the Sonogram picture update the information box in the under right corner. The info dialog (activated with scroll icon

With the home button the original help page will be shown again. The online help shows the `index.html` file in sonogram folder.

on the toolbar) let show detailed information. If the mouse is pressed on the Sonogram picture the yellow slider can an specific range in time-domain. Zoom in button on the toolbar zoom into this selected range. All zooms are stored in an internal list for later re-zooms to previous zooms. Zoom back to full signal show Sonogram generated first time opened the file. Playing button plays only the signal selected with the yellow slider. With F11-key the main window can maximized to full screen and the toolbar will be disabled.

5.2 Configuring the Sonogram with the Adjustment Dialog

Turn off the “automatically searching best window length“ let you select the window length manual. After selection the calculate button must be pressed again to update the spectrum. Most changes in the for the main window must be done manual recalculate button or with redraw button.

5.3 Analysing Windows

On the toolbar four analyse windows can activated.

- Fast Fourier window.
- Cepstrum window.
- Linear Prediction window.
- Waveform window.

All these four windows are updated immediately when mouse is moved over the Sonogram of the main window. Time point for the computations is the time point under the mouse. Is the time window from the mouse to the end of the signal longer than the time window selected for the signal, an red warn message appears in the upper right corner of the analyse window. All windows where closed with the main close button for the window.

5.4 3D Surface Plot

Pressing the 3D button on the toolbar begins generating of the 3D surface plot. If the Sonogram has

more then 20000 analyse pictures a warn message is shown and ask you for continue the surface plot generating. Plots with more than 20000 elements are very slow in interactive movement. With right mouse button the plot can cycled and with left button plot can moved. Zooming happened with middle mouse button (or on Windows systems by pressing right and left button together). Three plot types are available: point-cloud, line-grid and solid-surface. The background of the plot can be either monochrome colored or have a picture. The dimensions of the ground area is modifiable with the sliders for time and frequency. The data reduction feature is in beta phase and work not correct for all types of plots.

5.5 Working with SVG

Scalable Vector Graphic is an XML standard for vector graphics defined by the WWW Consortium. In comparison with the bitmap format SVG-XML files store the data in human readable format like html. In the export dialog colors and the size of the single specpoint can be selected. Exportation can be done in two different ways; pure SVG how defined in the W3C or with additional informations over the plot like filename, adjustment informations and sample informations. Pure SVG files can not re-imported into the program! To reimport of Sonograms SVG files simple open the file with the SVG file chooser. The time-domain based information for the amplitude representation on the bottom of the main window will be reconstructed from the spectrum.

6 Future Work

Wavelet transformation seems to be usefully for acoustical octave analyse and will be included in next time. An video-window which shows the current video parallel to the played or scrolled position seems easy to realize with java media framework. Sonogram should became a speech analyser function that work with ”Hidden Makrov Models”. At this point the speech analysis function is in width distance from development actual point

of development. Java source code for HMM algorithm is available.

A Musical Frequencies

In *c-dur* gamut musical frequencies are separated in twelve tones (*C C# D D# E F F# G G# A A# H*). Reference point is $f_a = 440Hz$. Other frequencies are calculated with $f_x = 2^{\frac{tone}{12}} f_a$ with *tone* = *difference between seeked tone and reference tone*. Every next octave has double frequency.

B About this Document

This document was build with L^AT_EX for Linux and Mac OS-X, t_EX version 1.0.

References

- [Roa96] Curtis Roads, “The Computer Music Tutorial”, MIT Press 1996.
- [Bri95] E. Oran Brigham, “Schnelle Fourier Transformation”, R. Oldenburg Verlag 1995.
- [Lau00] Christoph Lauer, Akkustische Signalanalyse/Synthese mit Wavelets im vergleich zur klassischen Fourieranalyse”, 2000.
- [Lau02] Christoph Lauer, onlinedocumentation for sonogram, 2002.