

**Spectro, A Spectrum Analysis Program**  
**by Perry R. Cook**  
**Stanford Center for Computer Research in Music and Acoustics**

Welcome to Spectro, a spectrum analysis application for the NeXT. Direct bugs/suggestions in large groups (save up a few before you mail) to Perry R. Cook [prc@ccrma.stanford.edu](mailto:prc@ccrma.stanford.edu). The program performs frequency transforms on sound data and displays the resultant spectrum. This is similar to passing light through a prism, which splits it into its component colors. To really use and understand all of the features of this program, you should take Julius Smith's sequence in digital signal processing for music offered at CCRMA. Optionally, you might read Xavier Serra's PhD. dissertation available from CCRMA.

### **Sound Capture**

The **Soundfile Capture** Window controls a single sound. You can record into the sound from the CODEC port by pressing the **Record** button. Recording is indicated by the sound meter above the sound view, and can be stopped by pressing the **Stop** button. After recording, the view shows the new sound. You can play the file by pressing **Play**, and the playing process can be aborted by pressing **Stop**.

You can load a previously generated sound file by typing the file name (with complete file path) into the **FileName:** field, and pressing the **Load** button or hitting carriage return in the **FileName:** field. **Save** saves the current sound to the file specified in **FileName:**. It is important to remember that the soundfile specified by **FileName:** is not necessarily the current sound. The current sound can be modified by recording, editing, or by loading from a file. If the current sound is modified by recording or editing, it will not match the file from which it was loaded.

The sound can be viewed in the sound view. If the sound is larger than the view, a scroll slider appears at the bottom of the view, and you can use it to scan across the sound. Pressing the **All** button causes the view to display the entire sound. **ZoomIn** and **ZoomOut** zooms the display, and **Select** zooms the display to the selected piece of the sound (see below).

By dragging over the sound in the window, a piece of the sound is selected and is displayed in grey. The entire sound can be selected by using the **Select All** command under **Edit** in the main menu. The selected portion can be copied or cut using the commands under **Edit** in the main menu, or by using the <command> key arguments. Once a piece is cut or copied, it can be pasted to any area within the sound, or even to another sound window. The sound segment is pasted wherever the cursor is located, and if a selection is made, the selection is replaced by the segment which is in the cut/copy buffer.

An important recommended operation is to remove any silence at the beginning and end. This makes signal processing easier later. Once you are happy with your captured sound, you can perform spectrum analysis on it.

## Spectrum Analysis

The defaults get you reasonable speech spectrum analysis at the CODEX rate of 8012 Hz. **Window Size** determines how much data is processed. **Window Time** tells you how long this is in seconds. **Zero Padding Factor** is a multiplier on the **Window Size** which determines the size of the transform. Frequency transforms in this program are restricted (for now) to powers of four, so if you pick a **Window Size** of 257, the **Zero Padding Factor** is automatically set to 3.984436 to yield a transform size of 1024. You can explicitly set the **Zero Padding Factor** to something greater than one for reasons which I will not explain here. **# Frames** is a positive integer which determines how many windows of data are processed. If **# Frames** is greater than one, the window is 'hopped' through the signal so that adjacent windows overlap by a factor of two. The **Hop Time** field tells you the time between adjacent analysis windows. **Total Time** tells you the length of the entire event that you can process. **Window Type** selects the window used for smoothing the data. I will discuss this no more except to say that if you want to see the effects of different windows, try a simple whistle at one frequency and process it with different windows.

Dragging over a section of the sound in the SoundView causes many of the fields to update. This is the spectrum analyzer calculating and displaying the time of the section that will be analyzed. **GO!** causes the region selected in the SoundView to be transformed to the frequency domain and displayed. If you select **Go On Select** in the **SoundFile Capture** window, the spectrum analyzer will recalculate and display the spectra each time you make a new selection in the sound view. The **Stop!** button makes the processing and display stop.

If **# Frames** is one or less (if you select a very small amount of the sound view), the frequency domain data is displayed in one frame in the **Real Time Spectrum Analyzer** window. Typing a negative number into the **dBRange** field changes the scale of the display. The plots are always referenced to 0 dB = maximum a sound can be, so by typing in the floor as a negative number, you set the range. Pointing to the display causes a cursor to appear and follow the mouse. This cursor causes the **Frequency of Cursor Position** and **Amplitude of Cursor Position** fields to display the frequency and amplitude of the spectrum at the cursor location. **Sinusoidal `Error' at Cursor** does nothing at the moment, but holds great promise for helping you interpret peaks in the spectrum in the future.

If **# Frames** is greater than one, the data is displayed in the **Waterfall Plot** view, with increasing time running upward. Don't be confused by the display, the plots are drawn from top to bottom, which is actually reverse time (it's easier to do hidden line removal this way, not your problem, mine). The slider at the right of the **Waterfall Plot** window allows you to select an individual spectrum for display

in the **Real Time Spectrum Analyzer** window. The time corresponding to the selected spectrum is printed in the **Current Time of Slider/Cursor** field. Changing the **FreqRange** of the **Real Time Spectrum Analyzer** view causes the **Waterfall Plot** to redraw all spectra. **Individual Plot Height** causes the individual plots in the waterfall view to be drawn scaled by that amount. If **Individual Plot Height** = 1, no plot will draw over another.

**Run** in the SoundFile Capture window (THIS BUTTON IS BROKEN IN VERSION 2.102 FOR NOW!!!!) will cause the **Real Time Spectrum Analyzer** window to calculate and display spectra continuously.

### Coming in the Future

1. Less bugs
2. More features, so new bugs
3. Real time spectrum analysis (Run mode).
4. DSP port mic support (I hope to support CODEX, 44.1 and 22.05 KHz mono or stereo for processing, and make the DSP do the conversion on the fly, summing both channels and doing sampling rate conversion to yield the 22.05 kHz file).
5. Help in interpreting features in the spectrum.