eMerge: toward a Knowledge-Based Spectral Editor

John Strawn
CCRMA, Department of Music
Stanford University
Stanford, California 94305
USA

Based on experience gained with text and soundfile editors, I have designed and implemented a spectral editing system with the following goals: easy to use; extendable; menu-driven; graphics-based.

To make the system easy to use, I decided to avoid user interaction based around a series of prompts, or to develop a unique command language parsed by the program. Bill Schottstaedt has implemented a window package for the Grinnel GMR--27 display driver; within this window package, I have implemented a menu package.

One kind of menu provides overall control. The features of the entire editing system (listed below) are summarized in one screenful of information. The user moves a cursor to the desired feature, which can then be invoked. When the user is done with that feature, control is returned to this main menu so the user can invoke other functions.

A second kind of menu allows the user to enter the name of a file containing a time-varying spectral representation of sound. Currently the editor handles outputs of the phase vocoder. The system is designed so that a time-varying spectral representation generated by any analysis technique can be accommodated; I am currently adding the linear predictor. Each new spectral file is entered into a linked list of files.

A third kind of menu is used to specify parameters for each feature of the editor. Each such menu is built from two columns: the left-hand column gives a short description of a parameter, with the value for that parameter in the right-hand column on the same line. Each menu is completely initialized to reasonable default values when first invoked. Using cursor-moving and text-editing commands closely modelled after CCRMA's in-house text editor E, the user can modify parameters and then invoke the function whose parameters are given in the menu. Any values modified are immediately checked to see whether they are reasonable. Each menu can be written out as a text file, providing a facile means to document work in progress. The same file (possibly edited with a text editor) can be read in again to change the menu values en masse. A separate set of such menus is maintained for each spectral file read in.

The features of the editor include: examining the original and/or modified spectra (on a channel-by-channel basis; in spectrographic and three-dimensional form); modifying the spectrum (data reduction algorithms; a cursor-based editor for modification by hand); and various utilities, such as a NEWS feature, and the capability to read in and write out modified spectra.

This system (more than 30,000 lines of code, not counting Schottstaedt's window package) is in use at CCRMA for psychoacoustic research and composition. It is written in SAIL, a superset of Algol, but has been designed from the outset with the goal of simplifying conversion to another high-level language, such as C; in particular, it is highly modularized, and each feature can be debugged individually.

The paper will conclude with a discussion of the experience derived from designing this editor, and of implications for the future, especially in the design of a knowledge-based sound editing system.