

# Editing Time-Varying Spectra\*

JOHN STRAWN\*\*

*The Droid Works, San Rafael, CA 94912, USA*

Time-varying spectral analysis is a useful tool for working with sound. Unfortunately a very large amount of data is involved. A menu-driven graphics-based editor for time-varying spectra has been implemented at the Center for Computer Research in Music and Acoustics (CCRMA), Stanford University. The features that have been developed for examining and modifying spectra are outlined, and suggestions are offered for the next generation of editors of this type.

## 0 INTRODUCTION

A number of signal-processing tools are available for the manipulation of sound. One class of such tools is time-varying spectral analysis, including the heterodyne filter [1]–[3], discrete short-time Fourier analysis (DSTFT, commonly called the phase vocoder) [4]–[15], the constant- $Q$  algorithm [16]–[20], linear prediction [21], the Wigner distribution [22], and the like. Analysis of a digital recording of sound using these techniques can result in a very large amount of data, perhaps on the order of 100 000 floating-point numbers for each second of the original. Because of memory and bus bandwidth limitations it is difficult to load this much data into memory and to display the data on a screen quickly with computer-based editors. The design of a flexible responsive editor is thus an important problem to solve.

There are certain historical precedents for an editor of this type. Backhaus [23] was perhaps the first to show the time-varying evolution of the amplitudes of groups of harmonics, with Risset and Mathews [24], [25] giving what was apparently the first modern example of this representation. Spectrographic plots of musical passages (such as those from the well-known Kay sonograph) are also common in the literature on the acoustics of musical instruments. In a rarely cited but extensive study Rösing [26] analyzed spectrographic plots of many orchestral passages and non-Western musical examples. (A summary is given in [27].) This line of work has been continued recently [28], [29].

LeCaine [30] printed the first spectrographic representation that we found of an individual musical tone; the first digital spectrographic plot we have found is in [31]. Some more extensive editing systems [32]–[35] have appeared, inspired in part by the requirements of speech research. However, none of these representations or systems offered the features needed for general acoustical research and musical composition.

From 1982 September to 1983 February we implemented a large editor for time-varying spectra at the Center for Computer Research in Music and Acoustics (CCRMA), Stanford University. (The CCRMA computer system at the time was based on a DEC PDP-10 mainframe computer, with real-time synthesis provided by the Samson Box digital synthesizer. Users worked on terminals featuring an ASCII keyboard and a monitor with moderate graphics capability.) Final modifications based on two years' experience with this editor were incorporated in 1985 January. The source code for the editor is more than 30 000 lines long. This editor has been designed to be easy to use. To this end it is modular, self-documenting, graphics based, and menu driven. The language SAIL [36], which includes ALGOL as a subset, was widely used at CCRMA, so it was used for implementation. (SAIL is now commercially available as MAINSAIL.) A few low-level routines were written in FAIL (an assembly language for the PDP-10); but the editor was designed to make conversion to another high-level language such as C as easy as possible. This paper will describe the features that were developed, and then concludes with suggestions for future implementations of time-varying spectral editors.

The current implementation, called eMerge at CCRMA, is designed primarily for work with the DSTFT. (The name comes from "merge files," used to store the analysis data.) "Hooks" have already been

\* Presented at the 78th Convention of the Audio Engineering Society, Anaheim, CA, 1985 May 3–6; revised 1986 March 26 and August 7.

\*\* Now with S Systems, P.O. Box 623, San Rafael, CA 94915.

included in the software for adapting this editor to other analysis techniques such as those mentioned earlier. Fig. 1 gives an overview of the DSTFT. In effect an input signal  $x(t)$  is passed through a set of band-pass filters whose center frequencies are equally spaced from dc to one half the sample rate. In analyzing tones from musical instruments, one usually arranges the filters so that one harmonic falls into each pass-band. The real and imaginary outputs of the filters shown in Fig. 1 give a time-varying spectral representation of the signal. If the analysis outputs are fed directly to the synthesis part of the technique, the output  $y(t)$  is virtually identical to  $x(t)$ . The analysis data may be modified in various ways to produce tones that are more or less close to the original. Also, the real and imaginary outputs may be converted into time-varying amplitude and frequency terms. The DSTFT and related techniques have been used for several decades to analyze musical instruments, yielding results useful for psychoacoustic researchers [1], [7], [23], [24], [27], [37]–[43] as well as synthesizer manufacturers and the recording industry in general. The fact that this system is so useful makes it important to be able to examine and modify the analysis output.

The following sections discuss several formats for examining the time-varying spectra that have been developed. These options, and the means for controlling them, were developed as a result of using early versions of eMerge. As such, these features are closely matched to the actual needs of users editing spectra.

## 1 EXAMINING ONE HARMONIC

With the DSTFT one channel produces traces for both frequency and amplitude. Fig. 2 shows the fundamental of a 1-s clarinet tone played at A below middle C (220 Hz). The amplitude is shown in Fig. 2(a), dis-

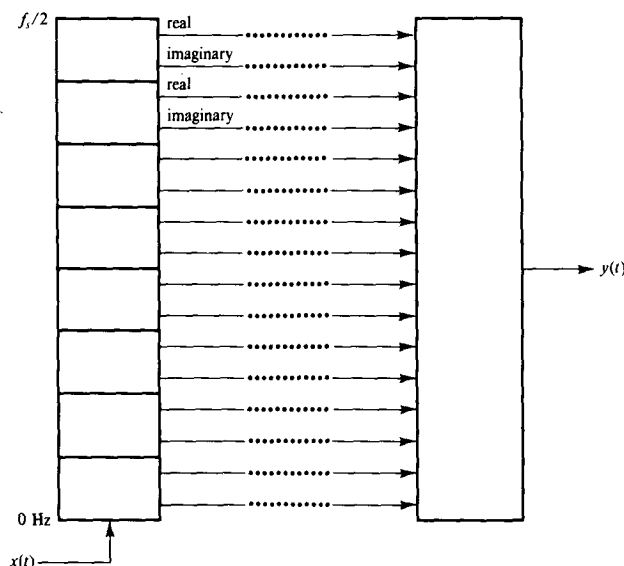


Fig. 1. The phase vocoder analysis stage (left) works like a set of band-pass filters spaced equally from 0 Hz to one half the sampling rate. If the data analyzed from  $x(t)$  are used for resynthesis (right), a signal  $y(t)$  results, which is nearly identical to the original. (Reprinted with permission from [10].)

played on an arbitrary linear scale (decibel plots are optionally available). The plot in Fig. 2(b) shows frequency in hertz on the ordinate. Time is the abscissa in both plots. (The frequency trace rises in Fig. 2(b) because a new note at a higher pitch is starting. This is also the reason why the amplitude plot does not return to 0 at the end of the note.)

Such plots cannot always be taken at face value. Consider the left-hand part of Fig. 2(b) with its wild variations. The output of the DSTFT “runs wild” when the amplitude of the passband filter’s output is extremely low.

To a certain extent this is an unfair example, since plots of the higher harmonics of actual tones are usually not so clean. Fig. 3 shows a typical example—the 25th harmonic from the same tone. Note that the amplitude is now much smaller, and the frequency trace contains extraneous fluctuations during the note as well as in the attack and decay. Our work with these tones [27], [43], [44], and that of Grey [41] and Charbonneau [37], lead to the conclusion that one can trust the general outline of the amplitude and frequency traces in such plots, but the microstructure of the traces remains unimportant.

Examining an individual harmonic in this manner is useful when one wants to

- 1) Determine whether a signal exists at all for the given harmonic.
- 2) Find the overall shape of the amplitude and frequency traces.
- 3) See how the amplitude and frequency interact in a given harmonic. For this, it is useful to be able to display the two traces on top of each other.
- 4) Look at specific parts of the harmonic’s evolution.

Capabilities 3) and 4) deserve further discussion. Fig. 4 shows the attack portion of the note given in Fig. 3. The amplitude and frequency traces have been overlapped (with a resulting confusion in the tick marks and labels on the ordinate). One possibly useful insight gained from examining Fig. 4 is that the frequency trace does not stabilize until  $t = 0.3$  s, although the amplitude trace becomes significant at about  $t = 0.22$  s.

In actual production work with real tones it is really necessary to be able to “zoom in” on parts of a harmonic in this manner. To give another example, Fig. 5 shows the attack of the eighth harmonic of a trumpet tone, also played at 220 Hz. One often finds characteristic blips in the attacks of brass instruments, as can be seen in Fig. 5(a). These blips may well occur not only in the first few harmonics, but even in the first 10 or 20 harmonics. (An example will be given later.) If such blips are omitted when one resynthesizes the note, then the result can be a tubby trumpet indeed [27].

Given the wild nature of the frequency traces during low-amplitude portions of the harmonic’s evolution, we found it advisable to include what we call an optional “squelch” on the frequency traces. Fig. 6 shows the plot of Fig. 3 with the frequency trace forced to a certain value (specified by the user) when the amplitude trace falls below a certain threshold (20 dB below the

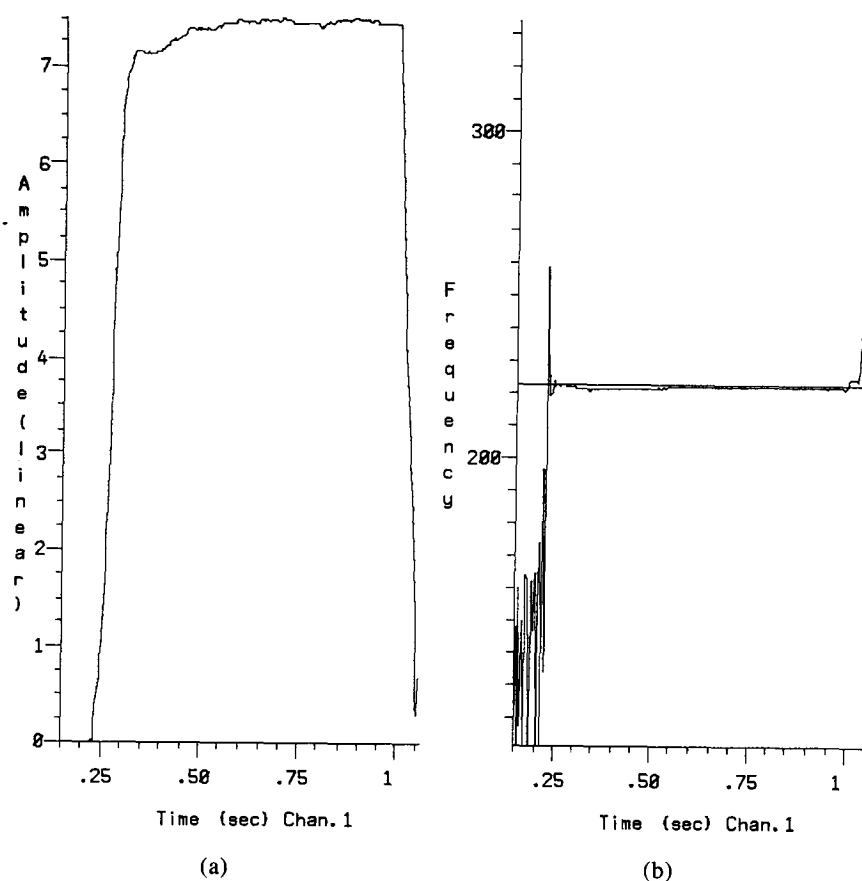


Fig. 2. Amplitude and frequency plots of the fundamental of a clarinet tone.

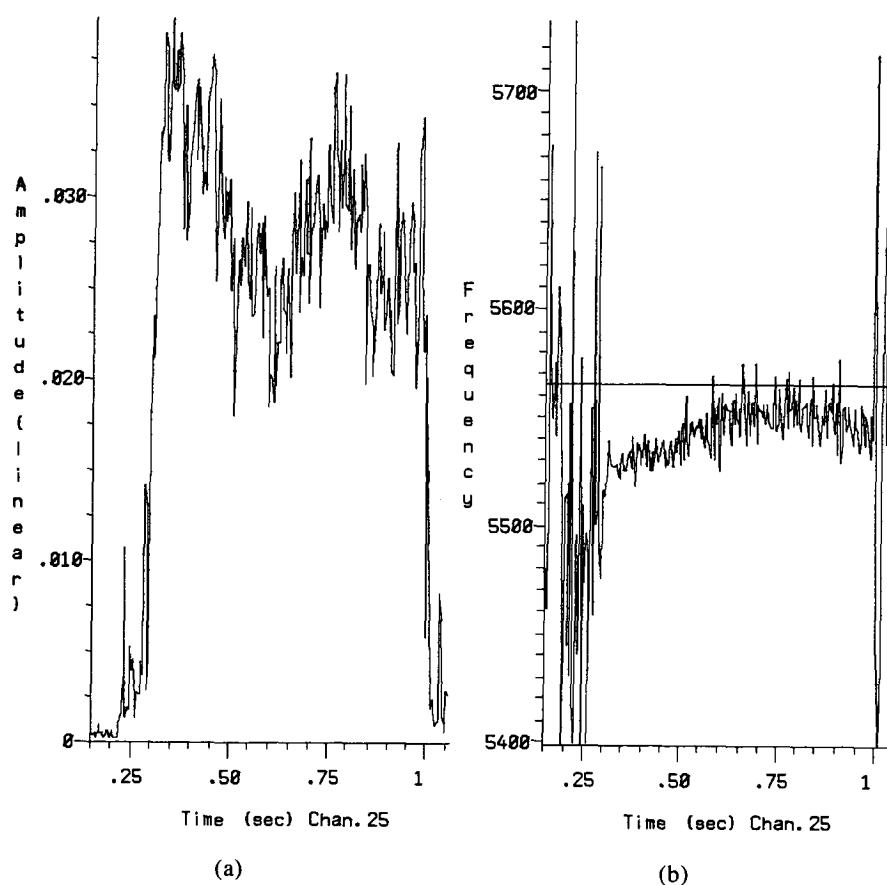


Fig. 3. Amplitude and frequency plots of the 25th harmonic of the same clarinet tone shown in Fig. 2.

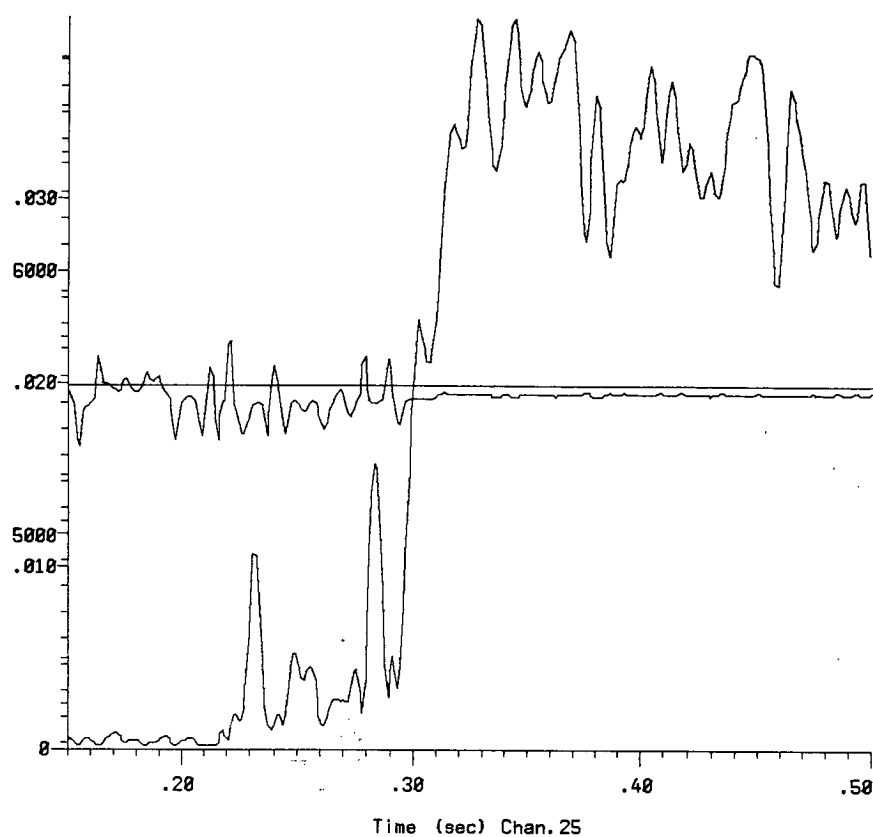


Fig. 4. Attack of the 25th harmonic of the clarinet tone shown in Fig. 2, with amplitude and frequency plots overlapped.

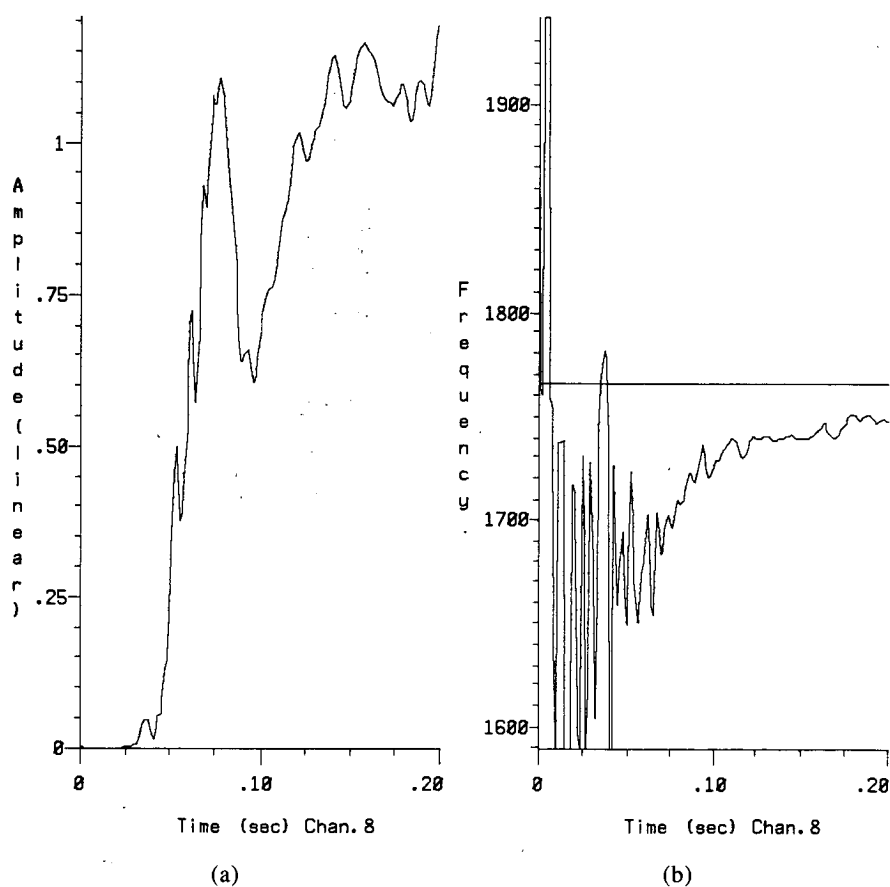


Fig. 5. Amplitude and frequency plots of the eighth harmonic of a trumpet tone. Notice the blips in the attack of the amplitude trace.

harmonic's maximum amplitude in this case). In this manner one can "clean up" the plots. This is often especially useful in the three-dimensional plots discussed next.

## 2 THREE-DIMENSIONAL PLOTS

Often one needs to examine more than one harmonic at a time. Fig. 7 is a three-dimensional representation of the first eight harmonics of a violin tone played at 220 Hz. Again, amplitude (this time on a decibel scale relative to the harmonic with the largest amplitude) is on the left, and frequency is on the right. (Frequency values in hertz are given only for the fundamental.) The fundamental is at the rear of both plots. Time (in seconds) is given on the abscissa for the eighth harmonic.

Actually several different kinds of three-dimensional plots are useful, depending on the applications. Fig. 8 shows another kind of representation, in which the channel number (corresponding to frequency) is the abscissa, and time rises from the bottom of the plot to the top. The amplitude of each channel is shown in decibels; each wavy horizontal line connects all of the channels at one slice in time. (This was how we generated the plot on the dust jacket of [45].) Fig. 8 shows the end of a note played at A220, followed by the (tongued) attack of the next higher C#. By varying the time window, the number of channels, and the decibel

range, one can cause the peaks and troughs to stand out more or less. If the number of analysis channels is high enough (several channels per harmonic), one can follow the frequency evolution of harmonic by tracing along the peaks in the plot from front to back, although this effect is subtle in Fig. 8. (Similar plots have been used to good effect by others; see [20], [39], [40], [46].)

One might want to have the choice of including hidden-line algorithms to clean up the plot. We tested such a facility but did not find it useful for production work. In general, one needs to be able to see detail which would otherwise be obscured by foreground harmonics in a "hidden-line" plot.

On the other hand, it is useful and sometimes amusing to be able to move the three-dimensional plot around interactively. Fig. 9 gives six views of the attacks of the first 16 harmonics of the same trumpet tone shown in Fig. 5. In Fig. 9(a) the viewer is looking directly "through" the harmonics from front to back. Such a plot shows that the harmonics are divided into two major groups with different attack characteristics. The front of the plot is moved down in Fig. 9(b); the fundamental is at the back. One sees here that the initial analysis is oversimplified: the top seven harmonics form one group, the bottom five harmonics are in another group, and the middle four harmonics form a transition region. This figure is rotated to the right in Fig. 9(c); the blips mentioned earlier are now easy to see. For

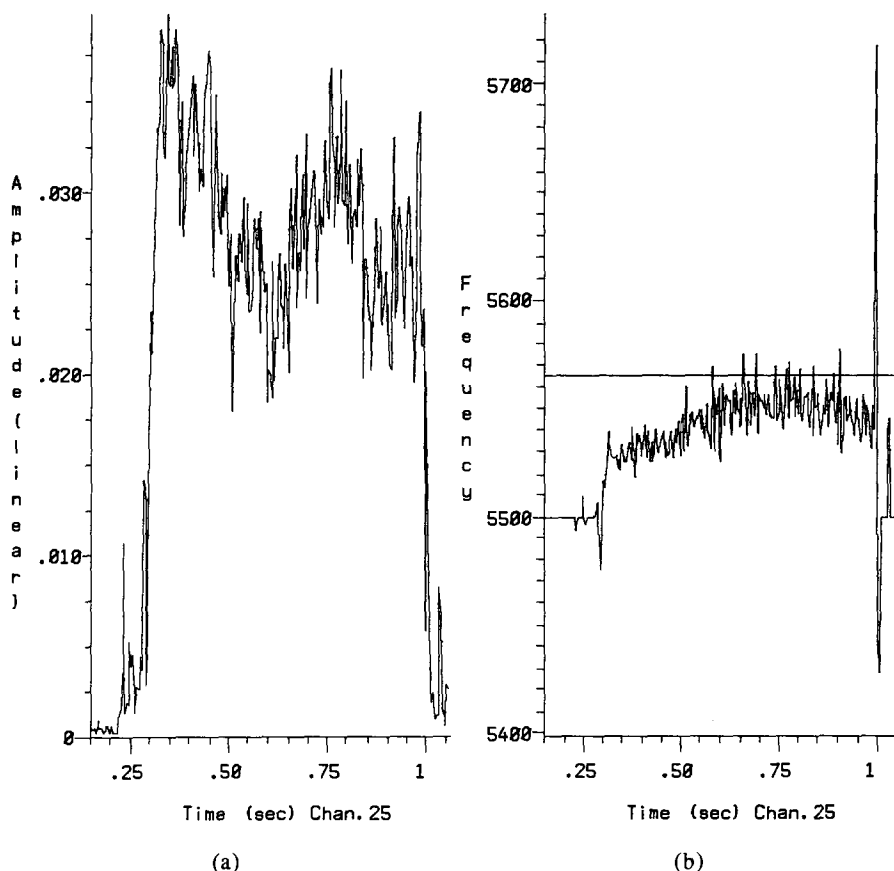


Fig. 6. The frequency trace of Fig. 3 has been squelched.

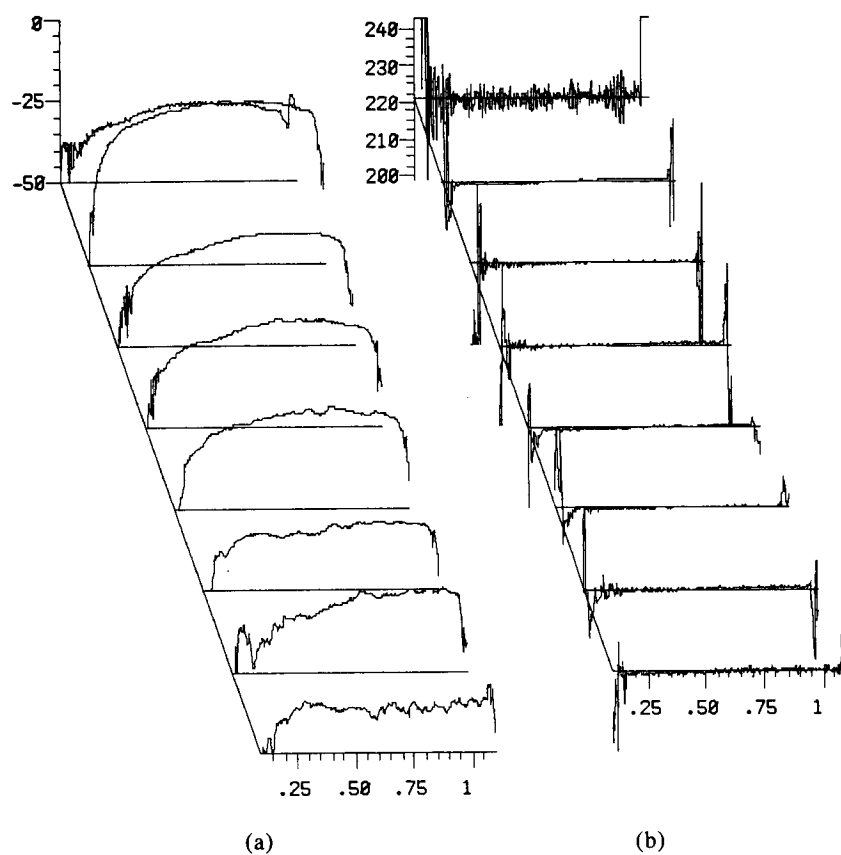


Fig. 7. Three-dimensional representation of amplitude and frequency plots for the first eight harmonics of a violin tone.

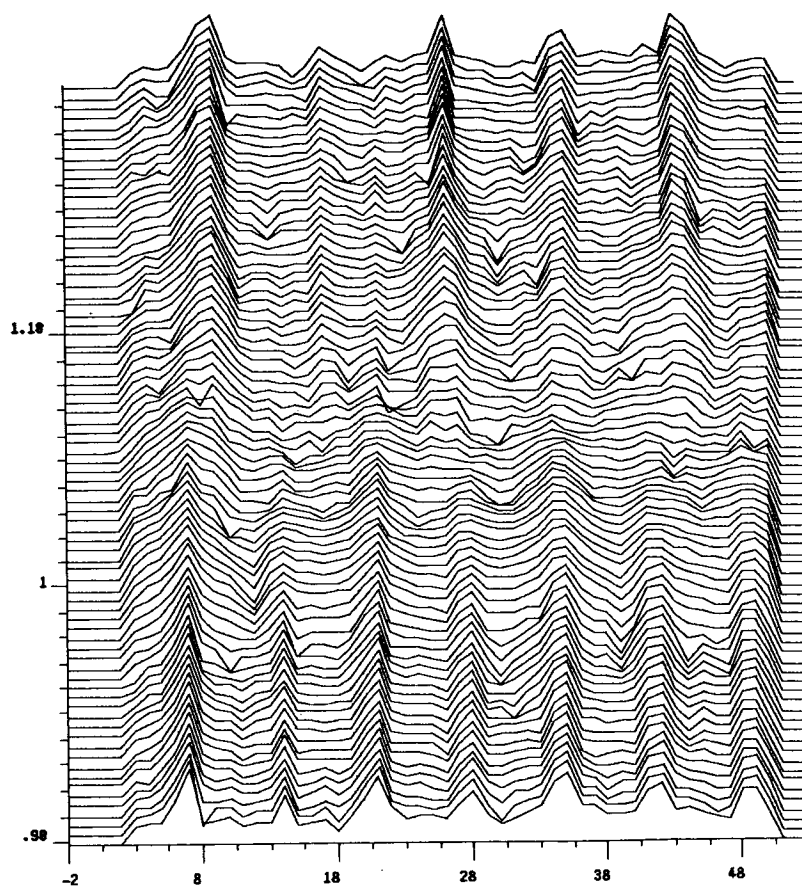


Fig. 8. Three-dimensional representation of the end of one clarinet note and the beginning of another.

Fig. 9(d) the plot has been turned around completely; here we see the attacks of the harmonics, looking back toward the beginning of the note from the steady state. The fundamental is on the right, and it is now easier to see that the fundamental is by no means the loudest of the harmonics. Indeed the hump in the amplitudes of the middle harmonics would suggest some sort of formant behavior. Fig. 9(e) shows the same spectrum from below (the fundamental is at the top), and the view directly from above the spectrum is given in Fig. 9(f).

In general, three-dimensional plots are useful for finding differences among different channels. They also provide a quick way for finding where significant spectral components occur. In some cases they are even useful for isolating problem spots. For example, there may be some noise in the preattack portion of a harmonic, which might show up as a squeak when the note is resynthesized; the first two harmonics shown in Fig. 9 are one such danger spot. Using a three-

dimensional plot of the attack, it is possible to find such places quickly. Other uses of these three-dimensional plots are discussed in [27], [44].

### 3 SPECTROGRAPHIC PLOT

The spectrographic form of a spectral plot (Fig. 10) is inspired by the spectrographic plots developed for speech work. In such a plot each harmonic is represented by a horizontal bar. The vertical position of the bar gives the frequency of the harmonic; the thickness of the bar gives the harmonic's amplitude. In this plot the horizontal bar is made up of a series of vertical lines, showing the values of the harmonic at discrete points in time. Spectrographic plots have proven more useful for educational purposes [47] than in research projects. For example, the plot of Fig. 10 shows the characteristic spectrum of the clarinet quite clearly, that is, the lower even-numbered harmonics are damped. Still, without a facile zoom feature it is difficult to

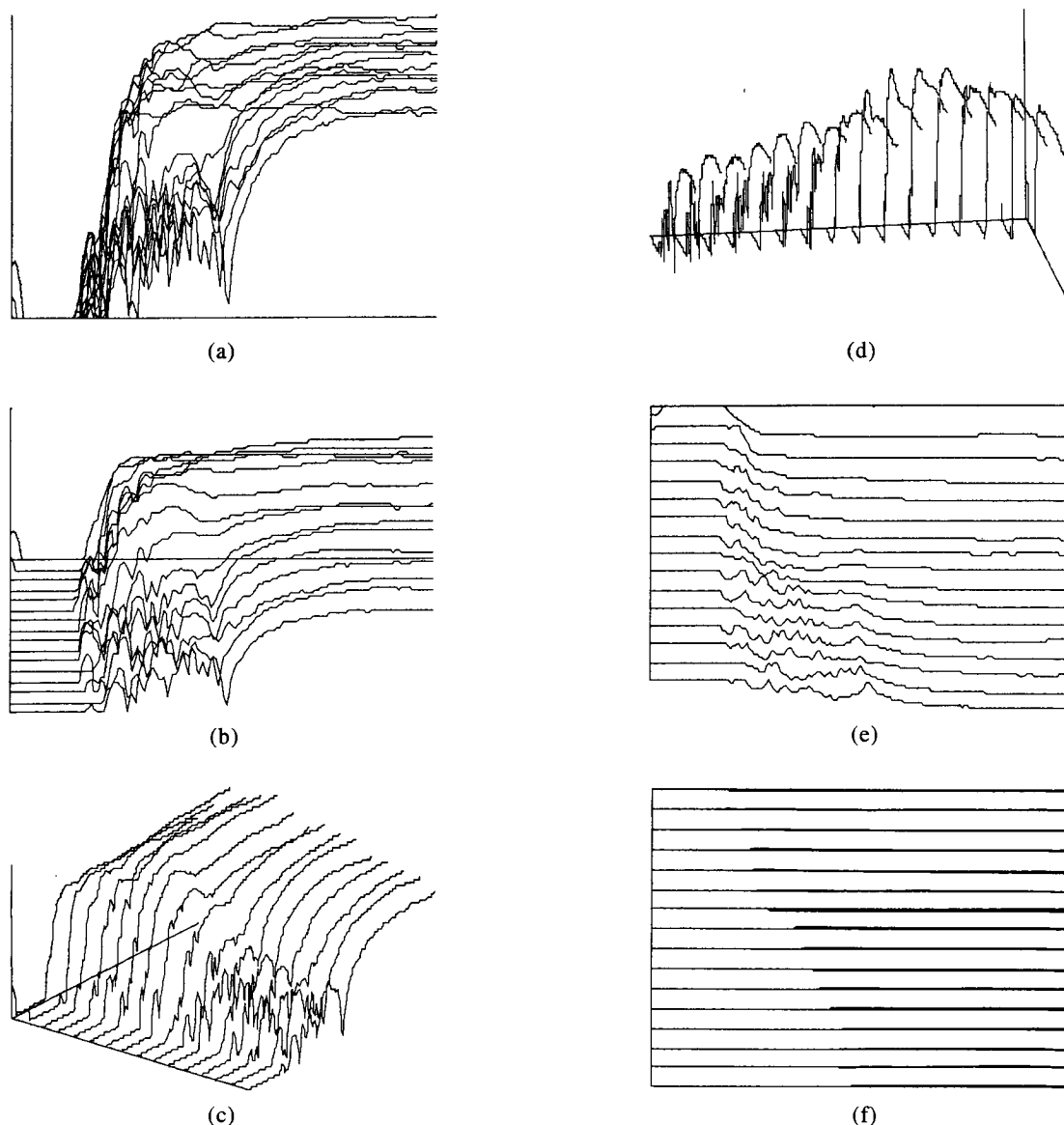


Fig. 9. Amplitude traces of the first 16 harmonics of the attack of a trumpet tone, seen from various angles.

glean detailed information from this form of plot. More plots of this kind are given in [26], [28], [29], [46]–[48].

#### 4 LINE-SEGMENT APPROXIMATIONS

Over the past few decades much research has been conducted into synthesizing more or less high-quality musical tones with line-segment approximations [27], [37], [41], [43]. In this manner, data reduction on the order of 20:1 through 50:1 or so can be achieved. In general the tones synthesized with line-segment approximations are difficult or impossible to distinguish from the original recordings, provided enough care is taken with the line-segment approximations. Such tones are useful in both commercial and research applications. In the commercial world, reducing the amount of data implies that a synthesizer with a given amount of computing power can synthesize more notes at the same time, or perhaps the same number of notes but with more complexity in each note. For research, reducing the number of points obviously reduces the amount of data to be manipulated. This makes it more convenient for the researcher to modify sounds in a controlled manner.

The spectral editor eMerge contains a sophisticated facility for creating and editing line-segment approximations. Fig. 11 shows an intermediate stage in editing the amplitude function shown in Fig. 3. The basic idea is to capture the outline of the amplitude trace with a

relatively small number of line segments. (Sometimes detail must still be captured in order to re-create a high-fidelity tone, as in the case of the trumpet blips shown in Fig. 5.) The cursor, denoted by an asterisk, gives the position where a point is about to be added. In general one moves the cursor around and adds or deletes points as necessary. But to be more specific, we added many features in the course of working with this editor, including the ability to

- 1) Move the cursor short or long distances in various directions:
- 2) Return the cursor to its previous position
- 3) Step along the original function
- 4) Step along the approximation, which is displayed on top of the original
- 5) Save the approximation, or restore a previous state of the approximation
- 6) Abort all changes
- 7) Print out the values of the original or approximation
- 8) Assign a specific time (in seconds), frequency (in hertz), or amplitude (decibels or linear) to a given point
- 9) Zoom in or out for fine or coarse editing
- 10) Optionally erase the axes, the original, or the approximation.

A short discussion of these features may prove useful. It is helpful to be able to return to the previous cursor position because one inevitably makes mistakes in typing cursor commands, some of which are easier to undo than others; simply returning to the previous position is often the best fix. Sometimes one needs to see the

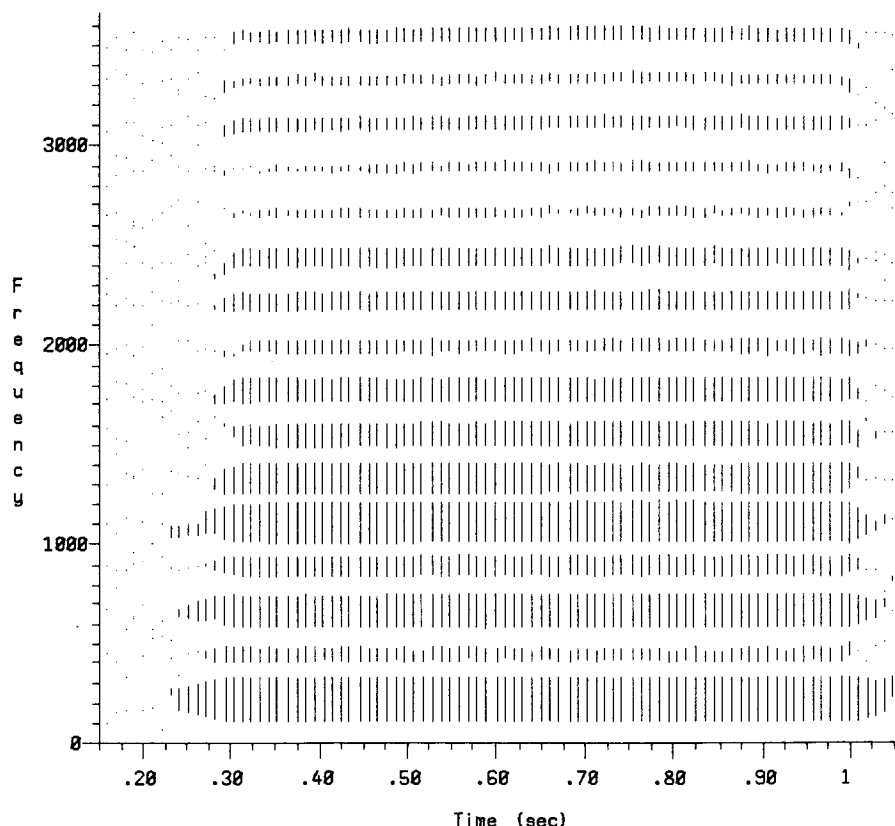


Fig. 10. Spectrographic representation of the first 16 harmonics of a clarinet tone.



values of the approximation in print because one wants to make sure one believes what one sees on the screen. Many other situations arise in which this is a handy facility. It is sometimes useful to erase the original because it is difficult to see the approximation, or vice versa; the attack in Fig. 11 is a case in point. Sometimes the axes need to be erased because they obscure the approximation or the original.

In adding points, it proved useful to have a "flexible rubber band" feature. Fig. 12 gives an example. As point A moves, the two lines emanating from it move with it. These lines show what the approximation will look like once point A is dropped into place. The nearly horizontal line above A shows what the approximation will look like without any point A. In this manner, one can align the slopes of lines in the approximation with the slopes of the lines (or outlines) of the original.

It is tedious to create such line-segment approximations by hand. We examined a number of algorithms from the literature on pattern recognition [49]. The Pavlidis split/merge algorithm discussed in [49] proved to be the most useful. Briefly, the user supplies an error threshold and chooses one of several methods for calculating error. The algorithm creates a line-segment approximation in which the error for each segment as calculated by the specified method is smaller than the given threshold. The results may be further cleaned up by hand using the editor just described. It has proven convenient to have the Pavlidis algorithm create initial line-segment approximations for all of the (several dozen) harmonics of a note at one time, then to edit

them later. This results in a substantial savings in console time for the user. More details are given in [27], [43].

We cannot emphasize enough the danger in designing editing systems around "clean" traces, such as those given in Fig. 2. In actual production work one deals most often with "ugly" cases, like that in Fig. 11. The system must be capable of letting the user make quick judgments about which aspects of the amplitude curve to include, and which to omit. Consider that a given tone may have 50 or so useful harmonics; even on a well-designed system with quick response time, one can easily spend 2 or 3 min on each amplitude function, resulting in a total console time of an hour or so just to make one note.

The spectral editor currently has room for three sets of line-segment approximations: the set as edited by hand, the set created automatically with the Pavlidis algorithm, and another, optional set. The cursor-based editor always writes into "its" set of functions, and the output of the Pavlidis algorithm always goes to "its" set. Each set of line-segment approximations can be written out to a file, or the set can be filled with functions from a file. (As an added feature of this facility, the points of the original analysis can also be written out to a text file of the same format, which is often useful in debugging the analysis programs.) As these sets are read in or written out to a file, operations can be performed on individual functions or on the entire set of functions. Such operations include scaling the values and/or the times of the approximation, or splicing two

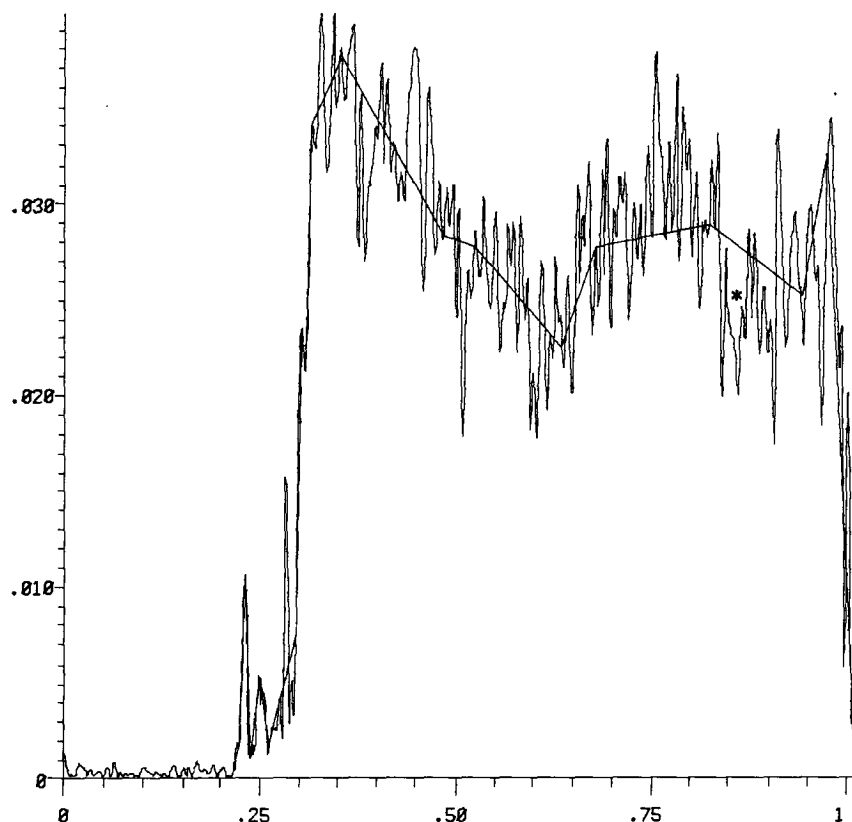


Fig. 11. Editing the amplitude of the 25th harmonic of a clarinet tone.

sets of line-segment approximations. In this way, one can operate on the functions at a meta level, which more closely approximates intuitive operations like tape splicing or filtering.

This brings up the database management question. Recall that a given note may have 50 or so harmonics. As stated earlier, each harmonic has a time-varying function for both amplitude and frequency. There can thus be as many as 100 or so such line-segment approximations for a note. It is important for the editor to keep track automatically of whether a function approximates a frequency or an amplitude trace, and what the channel number is. A simple data structure has been developed for this task. The functions are named A1, A2, A3, . . . , F1, F2, F3, . . . , and the spectral editor recognizes these names as the functions are read from a file.

The line-segment approximations may be viewed on a harmonic-by-harmonic basis, in a three-dimensional plot, or spectrographically. Fig. 13 shows line-segment approximations of the clarinet waveform given in Fig. 10. Here the approximation is outlined. Our experience shows that it is useful to be able to view the original or the approximation alone, or both together; Fig. 14 shows the approximation of Fig. 13 superimposed upon the original of Fig. 10. (Another example is given in [46].) Obviously the approximation does not capture all of the detail of the frequency traces in the attack; but resynthesis of the approximation shows that it is not necessary to do so.

## 5 SPECTRAL "AVERAGE"

Fig. 15 shows the discrete Fourier transform (DFT) of 100 ms from the steady state of the clarinet tone shown in Fig. 10 and elsewhere. (This plot was prepared using Rush's EDSND program [50] at CCRMA.) It is easy, using an appropriate editor, to pick off the peaks of the first three dozen or so harmonics. For the higher harmonics, life becomes more difficult. Again, this is a deceptively "clean" plot; experience shows that DFTs of musical instruments are not always so clean, especially for the higher harmonics.

Sometimes one needs a coarse measurement of the spectrum at every harmonic. To this end we developed a facility for averaging the amplitude and frequency outputs of the DSTFT over some user-specified duration. This approximates the behavior of the DFT, but has the advantage that a usable amplitude and frequency value for every harmonic can be easily obtained. Such an average is shown in Table 1 for the same section of the waveform shown in Fig. 15. (More examples of such tables are given in [27], [43].) The first column gives the harmonic number; the next two show the averaged amplitude, and the third shows the averaged frequency. The right-hand column contains values for what we term the "relative harmonicity" of the spectral component—how far it deviates from being an exact multiple of the fundamental. Using these values for additive synthesis of natural-sounding tones is discussed in [27], [43].

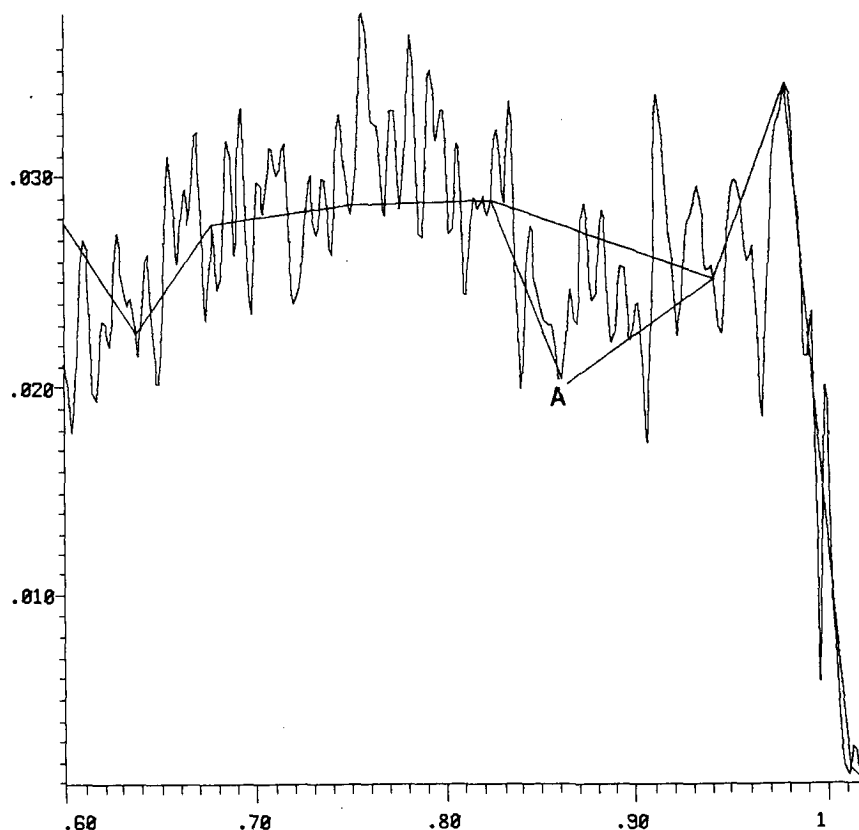


Fig. 12. Point A is about to be inserted into part of the waveform shown in Fig. 11.

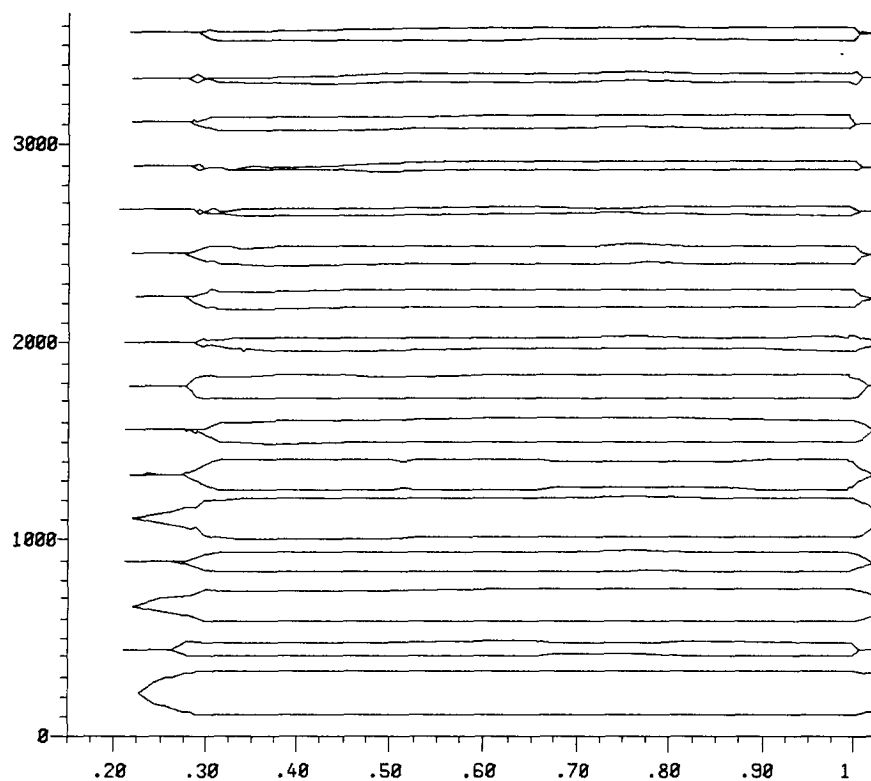


Fig. 13. Line-segment approximations of the tone shown in Fig. 10.

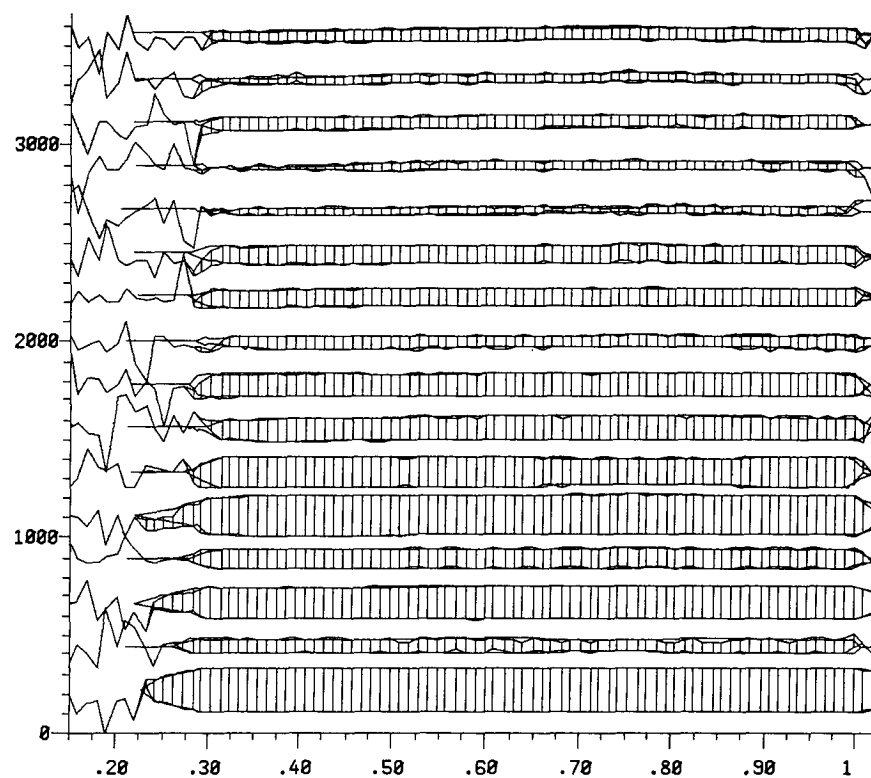


Fig. 14. Line-segment approximations of Fig. 13 shown superimposed upon the original analysis data of Fig. 10.

It needs to be emphasized here that the steady states of musical instruments are not exactly harmonic. The numbers in the right-hand column of Table 1 should make this clear. The spectral averaging feature of eMerge makes it possible to quickly derive reasonable frequency values for higher harmonics, even when the amplitudes of those harmonics are so small as to be close to the noise level of the recording.

## 6 ANALYSIS FILE INPUT

Sometimes one wishes to compare several notes of one instrument, or one note from several instruments, or even several analyses of the same note using different analysis parameters. The user may thus read in more than one analysis file at a time. Using an appropriate menu, the user has access to "bookkeeping" data about

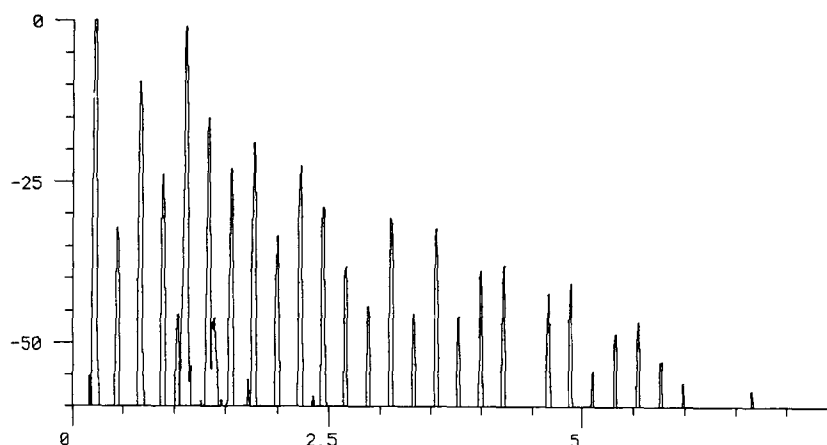


Fig. 15. Discrete Fourier transform of 100 ms of the steady state of a clarinet tone. Abscissa—frequency in hertz; ordinate—amplitude in decibels.

Table 1. Spectral average of the same tone used in Fig. 15.

Channel	Amplitude		Frequency (Hz)	freq <sub>n</sub>
	Linear	dB		freq <sub>1</sub> × n
1	0.3138	0.00	221.0724	1.00000
2	0.0075	-32.41	443.3353	1.00269
3	0.1032	-9.65	665.1712	1.00295
4	0.0190	-24.35	886.3823	1.00237
5	0.2745	-1.16	1108.2779	1.00264
6	0.0528	-15.47	1330.4116	1.00300
7	0.0215	-23.24	1551.3412	1.00248
8	0.0340	-19.30	1773.5556	1.00281
9	0.0064	-33.81	1995.4475	1.00291
10	0.0222	-22.97	2216.6563	1.00268
11	0.0110	-29.07	2438.9012	1.00292
12	0.0036	-38.73	2660.7559	1.00297
13	0.0019	-44.24	2881.5698	1.00265
14	0.0100	-29.85	3103.9618	1.00289
15	0.0018	-44.54	3326.0271	1.00300
16	0.0072	-32.73	3548.0196	1.00307
17	0.0016	-45.41	3768.9648	1.00286
18	0.0033	-39.31	3991.1366	1.00297
19	0.0041	-37.61	4213.1595	1.00304
20	0.0003	-59.06	4435.5923	1.00320
21	0.0023	-42.62	4656.4393	1.00300
22	0.0027	-41.11	4878.6143	1.00309
23	0.0006	-53.88	5099.9265	1.00300
24	0.0011	-49.01	5321.8991	1.00305
25	0.0014	-46.45	5543.6993	1.00306
26	0.0008	-51.46	5764.7940	1.00294
27	0.0004	-56.58	5987.0305	1.00303
28	0.0001	-65.46	6208.8262	1.00304
29	0.0002	-61.25	6431.2461	1.00314
30	0.0003	-57.94	6652.3293	1.00304
31	0.0002	-61.90	6873.6563	1.00298
32	0.0000	-71.38	7075.5868	1.00018
33	0.0002	-63.25	7318.4815	1.00317
34	0.0002	-62.69	7540.1416	1.00315

Note: This average spectrum was calculated over 0.1 s. The frequency of channel  $n$  is  $\text{freq}_n$ ;  $\text{freq}_1$  is the frequency of channel 1 (the fundamental).

each file, such as its history, the name of the file containing the recording analyzed, the sampling rate, number of channels, and so on.

## 7 USER INTERFACE

The main menu, shown in Fig. 16, allows the user to select any of the options available in the spectral editor. Each such option is controlled by a separate menu containing prompts for all the controls along with their current values. Fig. 17 shows the menu used to create the plot in Fig. 2.

In all of these menus user control has been designed so that the user works with the data in an intuitive fashion. Times are in seconds (or optionally in number of analysis points), frequencies are in hertz, amplitudes may be linear or in decibels. The user can zoom in and out of plots, specifying, say, smaller or larger decibel ranges. This version of eMerge "knows" about channel numbers in the DSTFT, so that the user can specify ranges of channels to be examined. The user may supply an optional label for the plot, and the whole screen may be optionally written out to a file. Other (more cryptic) lines control the placement of the picture on the screen.

Every menu has its own documentation, as well as type-checking on the control values. Thus in the first two lines of Fig. 17 the user may type only TRUE or FALSE (actually, T and F suffice); and in the third line only an integer number is accepted. (Floating-point numbers are, of course, truncated.)

Furthermore, each analysis file read into eMerge has its own set of menus, and the values in each menu are

remembered for each analysis file while the program is running. Fig. 18 shows the resulting data structure. The analysis files (labeled S in the figure) are connected as a linked list. Each S points to a complete set of menu values (M1, M2, M3, . . . ). Thus the user's work is remembered for each file so that the most recent settings for a given option are saved. It is also possible to write out a menu to a text file, which provides a useful way to log the progress of one's work, or to duplicate control settings in a later work session.

As shown in Fig. 18, each S also has its own triple set of line-segment approximations, labeled E (for edited by hand), A (for automatic approximation), and O (for other). In each set the two columns represent amplitude and frequency functions, and the rows represent harmonics.

## 8 OVERALL SOFTWARE DESIGN

To simplify the design of this program and debugging of future additions, each of the options listed in the main menu may be compiled and loaded in any arbitrary combination. This is accomplished with compile-time switches in the main program, which assemble a correct version of the main menu and require the correct load modules to be included. Thus one can work on a stripped-down version of the program for debugging, for the design of new modules, and for customizing the program for needs as they arise.

## 9 FOR THE FUTURE

The next generation of editors for time-varying spectra

```

Welcome to EMERGE, the Merge File Editor (Version 13.3)

Dealing with Merge files:
→ Get current input merge file
  Print directory of current input merge file
  Edit defaults for current merge file

Looking at data in current input .MF file:
  Show 1 channel
  3-d plot
  Spectrographic plot

Operating on data in current merge file:
  Pavlidis' Split/Merge Approximations
  EMERGE Function editor

Dealing with .FUN files:
  Read in .FUN file
  Write out .FUN file

EMERGE miscellany:
  News

α means to hold down the "ctrl" key
  while typing the next character
α? for help
αE to exit to next higher level
αG to execute option pointed at by cursor

```

Fig. 16. Main eMerge menu.

```

| Parameters for showing one channel from MF file UDP1:CA3TA.JMF [CLR,AWN]
|
Draw amplitude? (T or F) | TRUE
Draw frequency? (T or F) | TRUE

Amplitude plot:
  Amp x0 (in pixels, 0-511) | 0
  Amp y0 | 48
  Reference amplitude = maximum of channel? | TRUE
  If not, use this reference amplitude (=7.5116417) | 7.512
  Use dB for amp plot? (F = linear, T = dB) | FALSE
  dB Range | 50

Frequency plot:
  Freq x0 | 256
  Freq y0 | 48
  Squelch frequencies? (T or F) | FALSE
  Squelch frequencies to this frequency: | 242.7401100
  When the amplitude is this many dB down: | 50
  Freq. Channel Overlap (0.0 to 1.0) | 0.1

Parameters for both kinds of functions:
  Channel No (1:58) | 1
  Beg time | 0.2
  End time (2.09) | 1.05
  Also show Edited, Approximated, or "Other"
  functions? (Slot E, A, or O; N or F = none) | N
  Show merge function? | TRUE
  If so, draw it with lines? (F = dots) | TRUE
  Window Width (in pixels) | 256
  Window Height | 428
  Label |
  Include label (T or F) | TRUE
  Plot File Name | UDP1:AESF2.PLT

```

Fig. 17. Parameters controlling the display of an individual harmonic. These values were used to create the plot shown in Fig. 2.

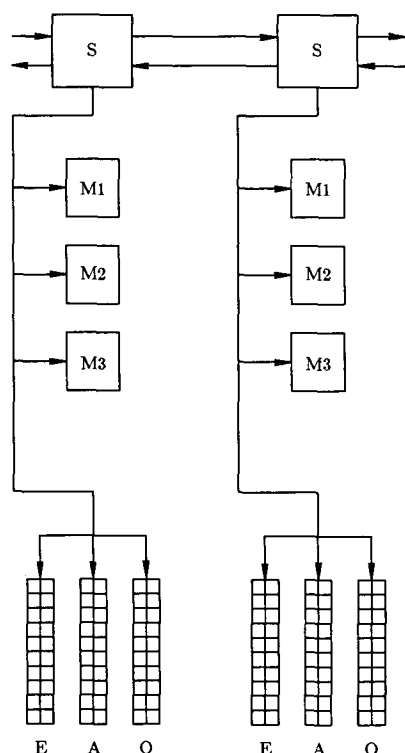


Fig. 18. Each analysis file (S) read into eMerge points to its own set of parameter menus (M1, M2, M3 . . .) as well as to one or more sets of line-segment approximations (E, A, O).

will be able to incorporate even more knowledge about the analysis data. Consequently, they will "know" about higher order timbral and acoustic labels such as longer, shorter, brighter, with a cleaner attack, and the like. With the appearance of graphics-based computer work stations with large memories and high-resolution screens, such as the Symbolics 3600, it will become even easier for the user to manipulate these large masses of data. An editing work station offering such capabilities will have useful applications in all aspects of sound editing. Even personal-computer-based systems will offer many of the capabilities outlined here [51], [52].

## 10 ACKNOWLEDGMENT

James A. Moorer and John M. Grey developed the merge file data structure for storing the outputs of time-varying spectral analysis in use at CCRMA. John W. Gordon contributed to the implementation of the DSTFT. Julius O. Smith III contributed two different software packages for making three-dimensional plots. Bill Schottstaedt wrote the window system and also contributed the axis-plotting software. Rick Taube and Xavier Serra served as guinea pigs during final debugging and offered helpful suggestions. This work was supported in part by the System Development Foundation.

## 11 REFERENCES

- [1] J. W. Beauchamp, "A Computer System for Time-Variant Harmonic Analysis and Synthesis of Musical Tones," in H. von Foerster and J. W. Beauchamp, Eds., *Music by Computers* (Wiley, New York, 1969), pp. 19–62.
- [2] J. A. Moorer, "The Heterodyne Filter as a Tool for Analysis of Transient Waveforms," Computer Science Dept., Stanford University, Rep. STAN-CS-73-379 (1973).
- [3] M. D. Freedman, "A Technique for Analysis of Musical Instrument Tones," Ph.D. dissertation, University of Illinois, Urbana (1965).
- [4] J. B. Allen, "Short Term Spectral Analysis, Synthesis, and Modification by Discrete Fourier Transform," *IEEE Proc. Acoustics, Speech, Signal Process.*, vol. ASSP-25, pp. 235–238 (1977).
- [5] J. B. Allen and L. R. Rabiner, "A Unified Approach to Short-Time Fourier Analysis and Synthesis," *Proc. IEEE*, vol. 65, pp. 1558–1564 (1977 Nov.).
- [6] R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing* (Prentice-Hall, Englewood Cliffs, NJ, 1983).
- [7] M. B. Dolson, "A Tracking Phase Vocoder and Its Use in the Analysis of Ensemble Sounds," Ph.D. dissertation, California Institute of Technology, Pasadena (1983).
- [8] J. L. Flanagan and R. M. Golden, "Phase Vocoder," *Bell Sys. Tech. J.*, vol. 45, pp. 1493–1509 (1966).
- [9] W. C. Gish, "Analysis and Synthesis of Musical Instrument Tones," presented at the 61st Convention of the Audio Engineering Society, *J. Audio Eng. Soc. (Abstracts)*, vol. 26, p. 898 (1978 Dec.), preprint 1410.
- [10] J. W. Gordon and J. Strawn, "An Introduction to the Phase Vocoder," in J. Strawn, Ed., *Digital Audio Signal Processing: An Anthology* (William Kaufman, Los Altos, CA, 1985), pp. 221–270.
- [11] S. Holtzman, "Non-Uniform Time-Scale Modification of Speech," M.Sc. and E.E. dissertation, Dept. of Electrical Engineering and Computer Science, M.I.T., Cambridge, MA (1980).
- [12] J. A. Moorer, "The Use of the Phase Vocoder in Computer Music Applications," *J. Audio Eng. Soc. (Engineering Reports)*, vol. 26, pp. 42–45 (1978 Jan./Feb.).
- [13] M. R. Portnoff, "Time-Scale Modification of Speech Based on Short-Time Fourier Analysis," Ph.D. dissertation, Dept. of Electrical Engineering and Computer Science, M.I.T., Cambridge, MA (1978).
- [14] M. R. Portnoff, "Time-Frequency Representation of Digital Signals and Systems Based on Short-Time Fourier Analysis," *IEEE Proc. Acoustics, Speech, Signal Process.*, vol. ASSP-28, pp. 55–102 (1980).
- [15] R. W. Schafer and L. R. Rabiner, "Design and Simulation of a Speech Analysis-Synthesis System Based on Short-Time Fourier Analysis," *IEEE Trans. Audio Electroacoust.*, vol. AU-21, pp. 165–174 (1973).
- [16] M. W. Callahan, "Acoustic Signal Processing Based on the Short-Time Spectrum," Ph.D. dissertation, Dept. of Computer Science, University of Utah (1976).
- [17] T. L. Petersen, "Acoustic Signal Processing in the Context of a Perceptual Model," Ph.D. dissertation, Dept. of Computer Science, University of Utah (1980).
- [18] T. L. Petersen and S. F. Boll, "Critical Band Analysis-Synthesis," *IEEE Proc. Acoustics, Speech, Signal Process.*, vol. ASSP-31, pp. 656–663 (1983).
- [19] G. W. Schwede, "An Algorithm and Architecture for Constant- $Q$  Spectrum Analysis," *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Process.* (1983 April), pp. 1384–1387.
- [20] J. P. Stautner, "Analysis and Synthesis of Music Using the Auditory Transform," M.Sc. thesis, Dept. of Electrical Engineering and Computer Science, M.I.T., Cambridge, MA (1983).
- [21] J. Makhoul, "Linear Prediction: A Tutorial Review," *Proc. IEEE*, vol. 63, pp. 561–580 (1975).
- [22] T. A. C. M. Claasen and W. F. G. Mecklenbräuker, "The Wigner Distribution—A Tool for Time-Frequency Analysis; Pt. I: Continuous-Time Signals; Pt. II: Discrete-Time Signals; Pt. III: Relations with Other Time-Frequency Signal Transformations," *Philips J. Research*, vol. 35, pp. 217–250, 276–300, 372–389 (1980).
- [23] H. Backhaus, "Über die Bedeutung der Ausgleichsvorgänge in der Akustik," *Z. techn. Physik*, vol. 13, no. 1, pp. 31–46 (1932).
- [24] J.-C. Risset, "Computer Study of Trumpet Tones," typewritten ms., Bell Laboratories, Murray Hill, NJ (1966); *J. Acoust. Soc. Am. (Abstracts)*, vol. 38, p. 912 (1965).
- [25] J.-C. Risset and M. V. Mathews, "Analysis of Musical Instrument Tones," *Physics Today*, vol. 22, no. 2, pp. 23–40 (1969).
- [26] H. Rösing, "Probleme und neue Wege der Analyse von Instrumenten- und Orchesterklängen," Ph.D. dissertation, University of Vienna, Austria (1967); (Verlag Notring der wissenschaftlichen Verbände Österreichs, Vienna, 1970).
- [27] J. Strawn, "Modeling Musical Transitions," Ph.D. dissertation, CCRMA/Department of Music, Stanford University, Dept. of Music Rep. STAN-M-26 (1985).
- [28] C. E. Potter and D. Teaney, "Sonic Transliteration Applied to Descriptive Music Notation," in H. S. Howe, Jr., Ed., *Proc. 1980 Int. Computer Music Conf.* (Computer Music Association, San Francisco, CA, 1981), pp. 138–144.
- [29] R. Cogan, *New Images of Musical Sound*. (Harvard University Press, Cambridge, MA, 1984).
- [30] H. LeCaine, "Electronic Music," *Proc. IRE*, pp. 457–478 (1956).
- [31] J. W. Beauchamp, "Time-Variant Spectra of Violin Tones," *J. Acoust. Soc. Am.*, vol. 56, pp. 995–1004 (1974).
- [32] L. H. Nakatani, "Computer-Aided Signal Handling for Speech Research," *J. Acoust. Soc. Am.*, vol. 61, pp. 1056–1062 (1977).
- [33] J. Strawn, *SYNTH Manual*, CCRMA on-line

documentation, Stanford University (1979).

[34] C. Roads, "A Report on SPIRE: An Interactive Audio Processing Environment," *Computer Music J.*, vol. 7, no. 2, pp. 70–74 (1983).

[35] D. J. G. Hirst, "Digital Sound Analysis and Synthesis Using the Short-Time Fourier Transform," M.A. thesis, Dept. of Music, University of LaTrobe, Australia (1985).

[36] J. Reiser, Ed., *SAIL*, Computer Science Dept., Stanford University, Rep. STAN-CS-76-574 (1976).

[37] G. Charbonneau, "Timbre and the Perceptual Effects of Three Types of Data Reduction," *Computer Music J.*, vol. 5, no. 2, pp. 10–19 (1981).

[38] M. D. Freedman, "A Method for Analyzing Musical Tones," *J. Audio Eng. Soc.*, vol. 16, pp. 419–425 (1968 Oct.).

[39] M. Piszczalski, "Spectral Surfaces from Performed Music, Pts. 1 and 2," *Computer Music J.*, vol. 3, no. 1, pp. 18–24; no. 3, pp. 25–27 (1979).

[40] M. Piszczalski, B. Galler, R. Bossemeyer, M. Hatamian, and F. Looft, "Performed Music: Analysis, Synthesis, and Display by Computer," *J. Audio Eng. Soc. (Engineering Reports)*, vol. 29, pp. 38–46 (1981 Jan./Feb.).

[41] J. M. Grey, "An Exploration of Musical Timbre," Ph.D. dissertation, Dept. of Psychology, Stanford University, Dept. of Music Rep. STAN-M-2 (1975).

[42] D. A. Luce, "Physical Correlates of Nonpercussive Musical Instrument Tones," Ph.D. dissertation, Dept. of Physics, M.I.T., Cambridge, MA (1963).

[43] J. Strawn, "Analysis and Synthesis of Musical Transitions Using the Discrete Short-Time Fourier Transform," *J. Audio Eng. Soc.*, vol. 35, pp. 3–14 (1987 Jan./Feb.).

[44] J. Strawn, "Orchestral Instruments: Analysis of Performed Transitions," *J. Audio Eng. Soc.*, vol.

34, pp. 867–880 (1986 Nov.).

[45] J. Strawn, Ed., *Digital Audio Signal Processing: An Anthology* (William Kaufman, Los Altos, CA, 1985).

[46] Front cover, *Computer Music J.*, vol. 8, no. 2 (1984).

[47] J. A. Moorer, J. M. Grey, J. Snell, and J. Strawn, "Lexicon of Analyzed Tones. Pt. 1: A Violin Tone; Pt. 2: Clarinet and Oboe Tones; Pt. 3: The Trumpet," *Computer Music J.*, vol. 1, no. 2, pp. 39–45; no. 3, pp. 12–29 (1977); vol. 2, no. 2, pp. 23–31 (1978).

[48] J. A. Moorer, "Signal Processing Aspects of Computer Music—A Survey," *Proc. IEEE*, vol. 65, pp. 1108–1137 (1977); reprinted in John Strawn, Ed., *Digital Audio Signal Processing: An Anthology* (William Kaufmann, Los Altos, CA, 1985), pp. 149–220.

[49] J. Strawn, "Approximation and Syntactic Analysis of Amplitude and Frequency Functions for Digital Sound Synthesis," *Computer Music J.*, vol. 4, no. 3, pp. 3–24 (1980).

[50] L. Rush, "The Tuning of Performed Music," presented at the 1982 Int. Computer Music Conf. (Venice, Italy), 1982.

[51] D. C. Massie, "The Emulator II Computer Music Environment," in B. Truax, Ed., *Proc. Int. Computer Music Conf.* (Computer Music Association, San Francisco, CA, 1985), pp. 111–18.

[52] L. Chouinard, L. N. Bélanger, and H. T. Huynh, "A Computer-Based Harmonic Analysis/Additive Synthesis System," in B. Truax, Ed., *Proc. Int. Computer Music Conf.* (Computer Music Association, San Francisco, CA, 1985), pp. 135–139.

---

The biography of John Strawn was published in the Jan./Feb. issue.