Chroma

Newsletter of the Australian Computer Music Association, Inc.
PO Box 4136 Melbourne University VIC 3052

Number 9
October 92
ISSN 1034-8271

In This Issue

Compact Disc of Electronic Music

The first ACMA CD *Machine Messages* has been released. It contains ten new works by ACMA members and gives a good indication of the wide range of musical interests among those active in the Association. We are currently negotiating a distribution deal through the Australian Music Centre in Sydney, but members can order a copy directly from ACMA for $20, which includes postage and packing. We are planning to release volume 2 by the middle of 1993, but need to receive pieces by the end of the year. If you are interested in having a piece released on volume 2, get in touch with Graeme Gerrard on (03) 344 4127 or write to him c/- ACMA, Inc. PO Box 4136 Melbourne University 3052.

ACMA AGM

Our Annual General Meeting is long overdue, but we have set the date for Monday 9 November, 8pm in the Heinze Room at the Conservatorium of Music, Melbourne Uni., Royal Pde, Parkville. The main business of the AGM is to table reports etc. and to elect a new committee. All positions must be vacated (President, Vice-President, Secretary, Treasurer) and we need new people with fresh ideas and time to do the work. We also need to look at projects for the coming year, whether it would be good to have an editor or sub-committee for *Chroma*, concert organisation etc., just to spread the work around and draw on people's specific interests and resources. To vote, nominate, second and stand for a place on the committee you only need to be a current, financial member. Due to work and domestic pressures Graeme Gerrard will be standing down as president this year after three years in the position.

NMA Conference

The New Music Australia Conference, organised by Sounds Australian, was held at the School of Music, Melbourne University 1-5 July. There were two sessions on electronic music research and applications. The papers themselves will eventually appear in the Conference proceedings, but as it is likely to be some time before the proceedings are published, some of the papers appear in this issue of Chroma.

Concerts

ACMA organised 2 concerts of electronic music for the Conference; at Elm St North Melbourne and at Melba Hall. Concert participants were: Chris Mann, Tom Fryer, Darren Steffen, Linda Ceff, Steve Adam and BIVA (Stuart Favilla, John Magill, Joanne Cannon, Pauls Sloss) on Friday and Jeff Pressing, Lawrence Harvey and Tim Kreger, Stuart Favilla and Michael Hewes, David Hirst and Graeme Gerrard, Amelia Barden with Michael Hewes and Chris Knowles on Saturday. Despite tight conditions and mountains of electronics the concerts went off well. The ABC recorded both concerts and have played excerpts on ABC-FM.

There was also an ACMA Concert for the Melbourne Fringe Festival on Saturday 5 September, with pieces by Graeme Gerrard, Warren Burt, Hope Csutoros, Chris Steller, Tom Fryer, Steve Adam and BIVA participants Stuart Favilla, Joanne Cannon and Adrian Sherriff.

An Approach to Algorithmic Composition with MAX

- Steve Adam

Since the middle of 1991, I have spent a great deal of time (coupled with a considerable amount of joy and frustration) developing patches using the MAX programming environment for the Macintosh. MAX is essentially a graphic object-oriented system enabling the user to create functional 'patches' which are comprised of a number of interconnected 'objects'. A wide variety of objects are available to the user and deal with MIDI and timing, events and numbers (both integer and floating point), lists, tables and data manipulation. It is also possible to create new objects in the Think C environment using object-oriented programming techniques.

While it is difficult to implement long term macro-structural/temporal control in the environment (though not impossible) and there is a limit to the number of processes that can operate simultaneously (due mainly to processor speed and event scheduling, and the memory requirements of complex patches), simpler program structures and data manipulations are easily achieved, even by users with little or no programming experience. It is also possible to have patches that contain or encapsulate other patches, with inlets and outlets communicating events to and from different patcher levels (subpatches). Furthermore, data structures can be modified and tested in real-time (as the program is running) and do not require recompilation. This is a particularly useful aspect, as it enables the rapid development and debugging of program operation.

MAX was initially developed by Miller Puckette at IRCAM, and much of the Macintosh version refined by David Zicarelli at Opcode Systems. The Macintosh version is a joint release by IRCAM and Opcode Systems. Recent developments have seen the extension of MAX into the audio processing domain through DSP control, and a new class of objects for real-time audio processing implemented on the NeXT version.

A full description of the application is beyond the scope of this article, however the interested reader may find more information in recent issues of CMJ and ICMC Proceedings.

Some projects this author has implemented with MAX include an Editor/Librarian for the Proteus 1 and 2, an input system for a video camera (a group project entailing the construction of proprietary hardware and the development of a specialized object), a number of algorithmic note generators and, most recently, the development of an accompaniment/improvisation system for live performance, which has been used in a number of performances/installations this year.

For the remainder of this article, I would like to focus discussion on the ongoing development of an accompaniment/improvisation system. While such a system has a considerable number of historical antecedents (not to mention current work in progress), it is hoped that the material provided here may be of some use to readers working along similar lines. The discussion assumes a working knowledge of MAX, however it should be relatively easy to follow if this is not the case.

The approach adopted in the implementation of the system was to create a number of modules, each of which perform specific tasks. From these basic modules, a performance patch is constructed, the behaviour of which is determined by the modules which it contains, and their interconnection. One of the most desirable modules to create was an input stage, or 'front end' which behaves in a similar manner to a garden variety sequencer. Essentially, the function of such a module is to encode a MIDI performance into its constituent parts and to tabulate the results (events). The events that are registered by the 'front end' are then made available to a variety of 'processes' (other modules) which can then access the recorded material and can perform functions determined by arbitrary conditions. A process here is defined as any modification of the material which is received, and is similar in concept to a signal modifier in sound synthesis parlance. On a faster machine, any reasonable number of processes may be active, allowing the realization of fairly dense orchestrations, even from single monophonic lines. The remainder of this article will deal most specifically with the design of a basic 'front end' for notes and as a consequence will only deal with MIDI note event parameters, and not with continuous controllers, aftertouch and so forth.

In order to have a process manipulate the recorded data, it is necessary to store such data so that each individual event (or event parameter) can be independently addressed in real-time. In this respect, the system differs from most sequencers, which usually give every recorded event a unique and absolute timestamp. While timestamping is useful for the accurate reproduction of note and real-time controller events in a linear time flow representation (such as a sequence), it is not necessary, and in some instances, can hinder the process of re-ordering note events.

Initially the relevant data (i.e.. MIDI note number, initial velocity, duration and time since last note-on received) must be presented in numerical form so as to enable subsequent storage. The Borax object in MAX provides information about note-on and off events that it receives when supplied with note and velocity information at its inlets, as well as keeping track of the number of notes currently active. This last feature is particularly useful when dealing with polyphonic input data. For each note-on event, Borax outputs a unique number (at the first outlet), which is then output again when the note is released. This two-staged process is necessary in order to capture all the details of the event and can be clarified by considering the sequence of events in the reception of a note. At the point in time in which the note-on message is received, we have knowledge of the MIDI note number, the initial velocity and the time since the last note-on (delta time). The duration of the note however, is only available once this note has been released (note-off). As a consequence, it is necessary to temporarily store the initial velocity and delta time values of a note-on event until its respective note-off is received, at which point all data relevant to the note-event can be committed to 'permanent'

storage. The MIDI note number does not require such treatment, as it is transmitted by Borax in both instances (note-on and off).
A further consideration for the temporary storage of the initial velocity and delta time values arises when dealing with extreme legato playing or polyphonic data. If a new note event is received before a previous note is released, the previous temporary values will be overwritten by those of the new note event. A solution to this problem is to have an array of temporary values, indexed by the voice number parameter available at the second outlet of the Borax object. This voice number assigns a numerical value to each new note event corresponding to the lowest available number. For example, in a monophonic line without note overlap the voice number will always be 1. If a note arrives while another is still active, the new note will be assigned a voice number of 2. This value is transmitted at both the commencement and conclusion of a note event.
The method of storage employed in the system is with the use of the table object. While alternatives are available, the table object simplifies the reading and writing of event parameters, and in addition, allows the user to view and edit data graphically as well as performing statistical analyses. One corresponding limitation in using the table object is that each addressable location can only store one value, as opposed to an array of values, and, in addition each table must have its size and display range values set in the Get Info dialog. In practice, this means that the representation of any note event will require a table for each recorded parameter.
Diagram 1 shows a basic event recording system as described above, while Diagram 2 reveals the inner structure of the 'tables' patch (p tables) located in Diagram 1.
In the example provided here, storage is configured in a circular buffer arrangement whereby only the last 300 notes will be 'remembered'(300 is an arbitrary figure; the table size can be set to any reasonably large value and, for initial experimentation, very small values are suggested). This value can be altered to suit the particular need of the experimenter by altering the value in the modulus or remainder object (% 299). The MIDI note number is processed by mod (%12) to derive the pitchclass value, and divided by 12 to obtain the octave value. Tables V and T are the temporary buffers referred to earlier, for velocity and delta time respectively. Upon the reception of a new note-event, the Pack object will combine the value received at the right inlet (velocity or delta time) with the voice number value (Borax outlet 2) and transmit these values as a 2 element list to the tables, at the address specified by the voice number value. When the note is released, the voice number is again output from Borax and the values in the corresponding address of tables v and t are transmitted to the tables patch, along with pitch, (pitchclass and octave) and duration. Each variable is written into its respective table at the address specified by the note number (+1) value, except for the delta time value which is written into the location of the previous note-event. This will generally make the design of 'Players' easier, as the arrival time of the next note may be 'known' in advance.
The next step in developing such a system is the creation of a basic 'Player' which can be defined as a process or series of processes which result in some kind of output (usually MIDI). Once this is accomplished and the basic system established, the real work of 'Player' development can commence. This is an important step, as the musical utility of the whole system will be determined by the 'intelligence' or otherwise of the Players developed. By way of introduction, Diagram 3 shows the patch for a very basic 'Player', which will play through the note buffer either forward or backwards, scale time, and can be set to commence at any location within the buffer. Diagram 4 shows the internals of the 'readtables' patch (p readtables) in Diagram 3.
Further development of 'Players' can take any number of directions, and will usually result in significantly more complex patches than the one shown here. My own experimentation has taken two distinct paths in approaching the design of Players. The first is toward a highly configurable generalized model, whose behaviour is controlled by a large number of parameters, menus and conditional tests. While there are advantages in this approach, particularly in the flexibility in the specification of Player behaviour, the net result tends towards a large and unwieldy collection of objects and subpatches which, through its complexity, becomes increasingly more difficult to modify or refine, and demands proportionally more processor time.
The second approach, and probably the more successful of the two is to incorporate the development of a range of independent modules or processes (as mentioned earlier) which are combined to form the architecture of a Player. By standardizing the input and output streams of such processes, their interconnection is more simply achieved and more easily modified. While such an approach is evident in the object-oriented nature of the Max program, the standardization of interprocess communication requires that the program be used as a kind of 'shell' for another level of program complexity, and considerable thought is also required in achieving a high degree of compatibility between the messages interpreted by different processes. In some cases, it is also useful to partially circumvent the 'box-wire-box' graphical programming approach in favour of more traditional methods (global arguments and variable names typed directly into objects (processes) through the extensive use of send# and receive# objects). This aids in the creation of complex patches without the use of 'wires' or patch leads, and is very useful where tracing signals between objects is a frequent occurrence and/or screen space is a limited resource.
In future issues (time and space permitting), further elaborations of the methods touched on here will be outlined in more detail. Suggestions, ideas or criticisms are welcomed and can be sent c/- Chroma or emailed to me : mussa@lure.latrobe.edu.au

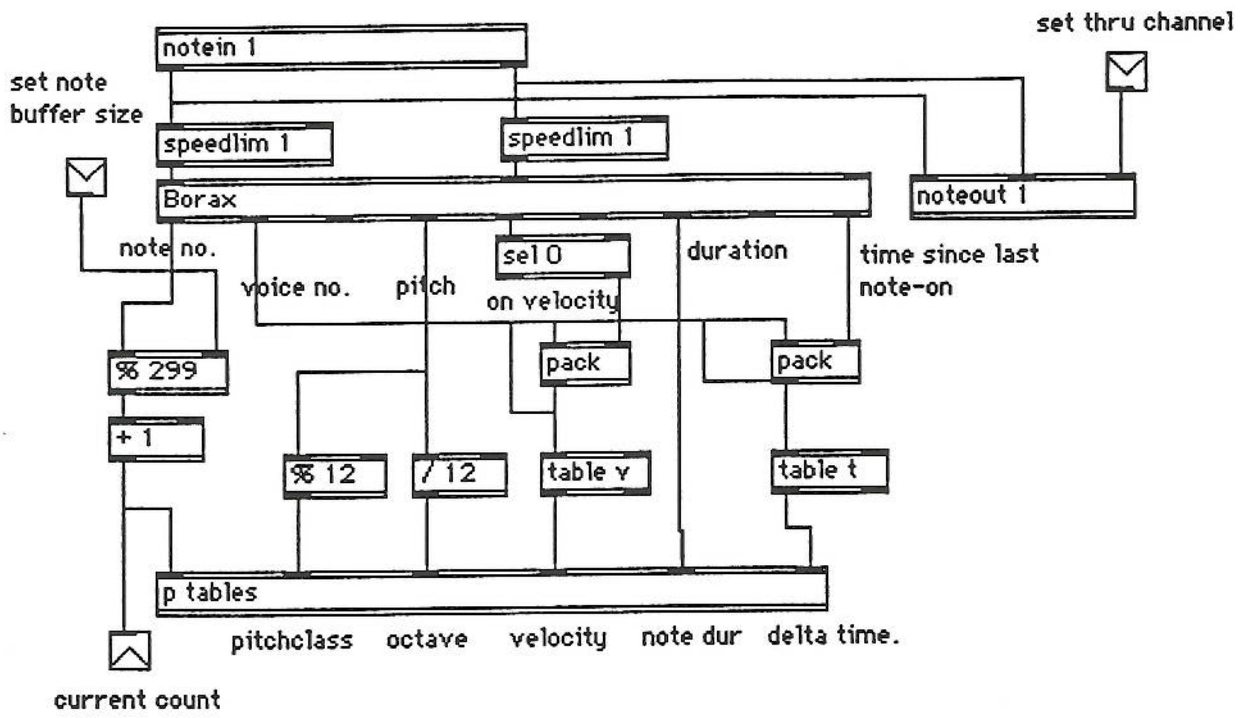

Diagram 1. The 'Front End' or Note-event recorder.

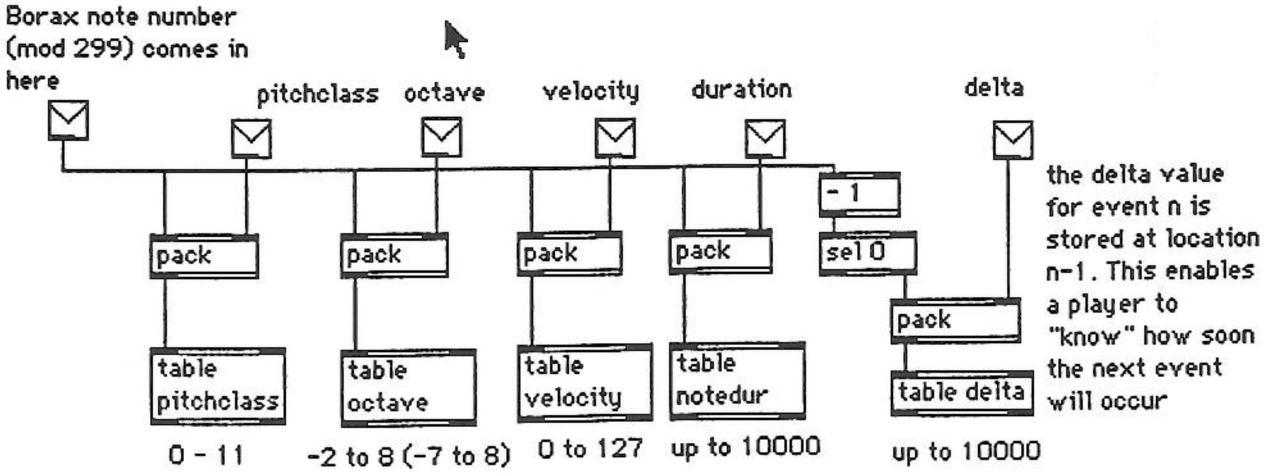

Diagram 2. The Internal structure of the 'tables' patch (p tables in Diag. 1.)
The values beneath each table object show the range that the table is likely to encounter.

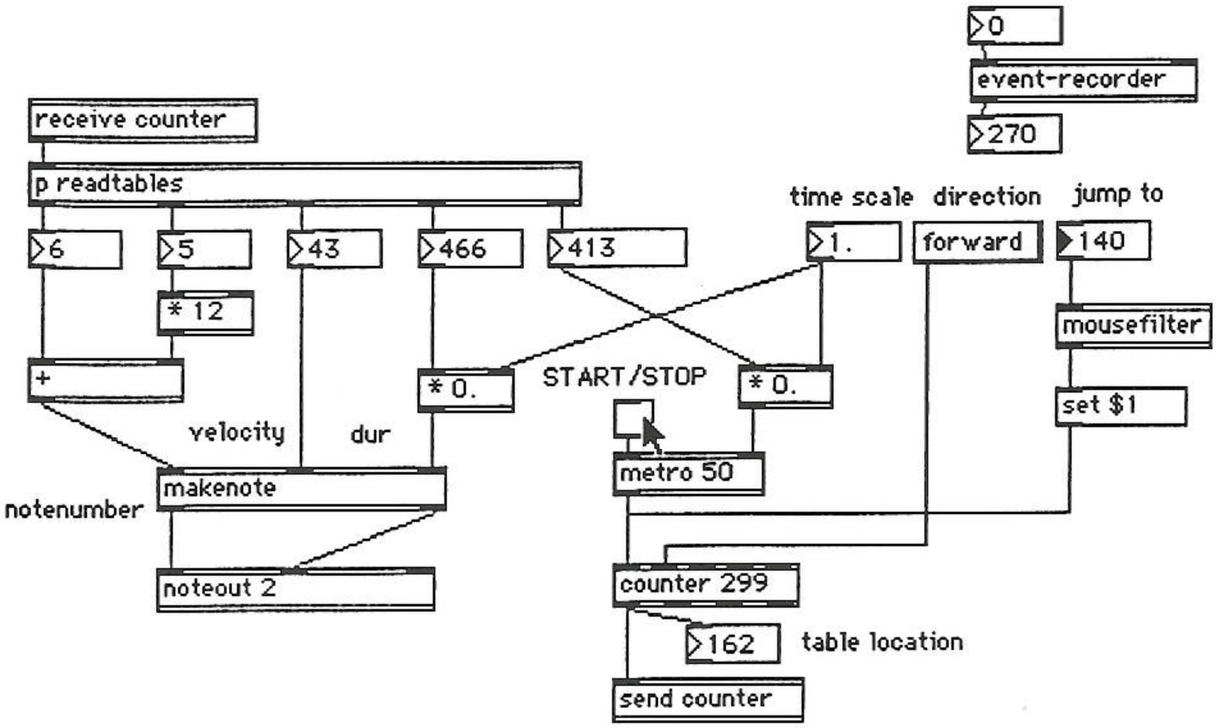

Diagram 3. The basic-player patch, which will play back the events recorded by the event-recorder which appears as an object in the top right hand corner.

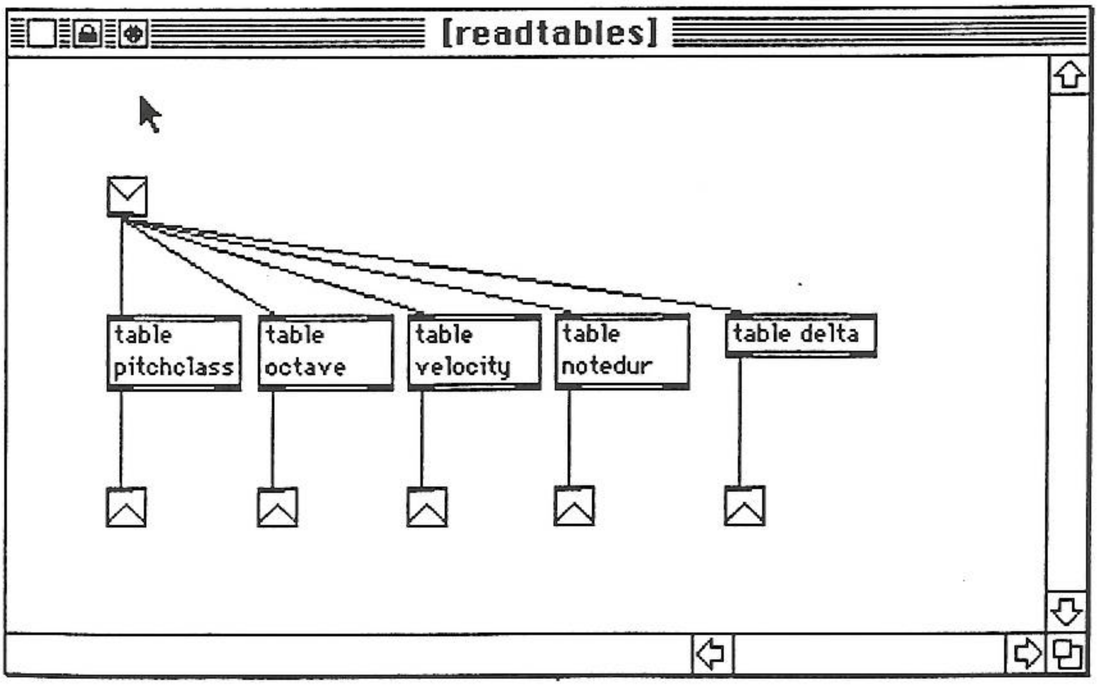

Diagram 4. The Internal structure of the 'readtables' patch. (p readtables in Diag. 1.)

Digital manipulation of cymbal timbres: sampling, spectrum analysis, and signal processing

- Robert Bell

The generation and manipulation of percussion timbres via electronic means has been researched and applied for many years, and is a significant aspect of the electronic music environment. The primary aim of most electronic percussion development to date has been to imitate acoustic percussion timbres, especially in commercial music, where sampling is exploited as an effective way to do this. Since 1991 I have been experimenting with electronically manipulating percussion timbres with a different objective; primarily, to create new percussion timbres, many with definite pitch, using a variety of Digital Signal Processing techniques. This work will culminate in a series of compositions, currently in progress, which exploit the melodic and harmonic potential of these new sounds, as well as the more traditional rhythmic and 'colouring' functions. This research is currently being conducted in the music department at La Trobe University, Victoria.

Examples of previous research - 1991 to date

Research began with experiments on several suspended cymbals, in which various extended performance, recording and sampling techniques were employed to increase the pitch definition of the instruments' timbres. Below are examples of experiments conducted by myself on an 18" crash cymbal. Recordings were made in the recording studio at La Trobe University, using a variety of microphones in a highly reflective room ambience. Sounds were recorded directly to digital tape and then transferred to hard disk in the Macintosh computer, using Digidesign's Sound Designer II. From there selected samples were analysed using Annalies* and modified using SDII's DSP facilities, primarily filtering. Sounds were then transferred to the Akai S1100 sampler for further editing, and for application in the MIDI environment (sequencing, performance, etc.).

Example 1: The cymbal was recorded with an AKG D160E omnidirectional microphone, with an ambient placement 2m from the cymbal and 2m off the floor. The cymbal was struck with the tip of a drumstick. No other treatment was imparted on the sound. Allowing for subjectivity, my personal observation was that this recording technique gave a very accurate reproduction of the cymbal's natural timbre, ideal for imitative purposes. A Fast Fourier Transform (FFT) of this sample shows a general picture of the instrument's broad frequency spectrum (Fig.1). Because of the obvious density of upper partials in the sound, there was little pitch definition and the sample was unsuitable for melodic applications in composition.

Example 2: A Sennheiser MD441 hypercardioid microphone was placed 5cm under the cymbal's edge, and the cymbal was struck as in example 1. The resulting timbre was significantly different to the previous example, with high frequency overtones severely attenuated and the fundamental (~85Hz) emphasised. An FFT of the sampled sound outlines this (Fig.2). This sample had quite clear pitch definition, although it contained a dissonant overtone (~365 Hz) which proved to be distracting when used melodically. This was solved by digitally filtering out the overtone with SDII's graphic equalizer. The final result was a timbre with strongly defined pitch that was particularly useful for bass applications in composition.

Example 3: The same microphone and placement were used here as in example 1. The cymbal was muted as I struck it, by holding its edge with my hand. High frequency overtones were attenuated by limiting the cymbal's vibration (Fig.3). This resulted again in a timbre with more defined pitch. A significant reduction in the cymbal's decay time (from ~15 secs to ~3 secs) also made the sound easy to control in rhythmic and melodic applications. Attenuation of dissonant overtones was performed again with the use of filtering. Using SDII's parametric equalizer, a notch filter cut ~3kHz with a 400Hz bandwidth, thereby attenuating a series of overtones spaced closely together. The result was a timbre with definite pitch that was useful in many melodic applications.

Other manipulation techniques were applied to almost all of the samples used in composition so far. Apart from filtering, one of the most widely used was modification of envelope, using the S1100's ADSR envelope generators. These were used to modify amplitude, pitch and filter envelopes for each sound. By adjusting rates of attack and decay in a sound, variation in articulation could be achieved. Furthermore, when envelopes were programmed to change with key velocity (across velocity zones), articulations were thus variable in relation to dynamics. This greatly enhanced the melodic expressiveness of the sounds. Other techniques used in the course of sound creation and composition included multisampling, digital mixing, transposition, stereo imaging and pan modulation, loop generation, using MIDI controllers, using digital delay as a rhythmic generator and so on.

* Spectrum Analysis program, where Fast Fourier Transforms of Sound Designer II files may be created. Program written by David Hirst and Thomas Stainsby.

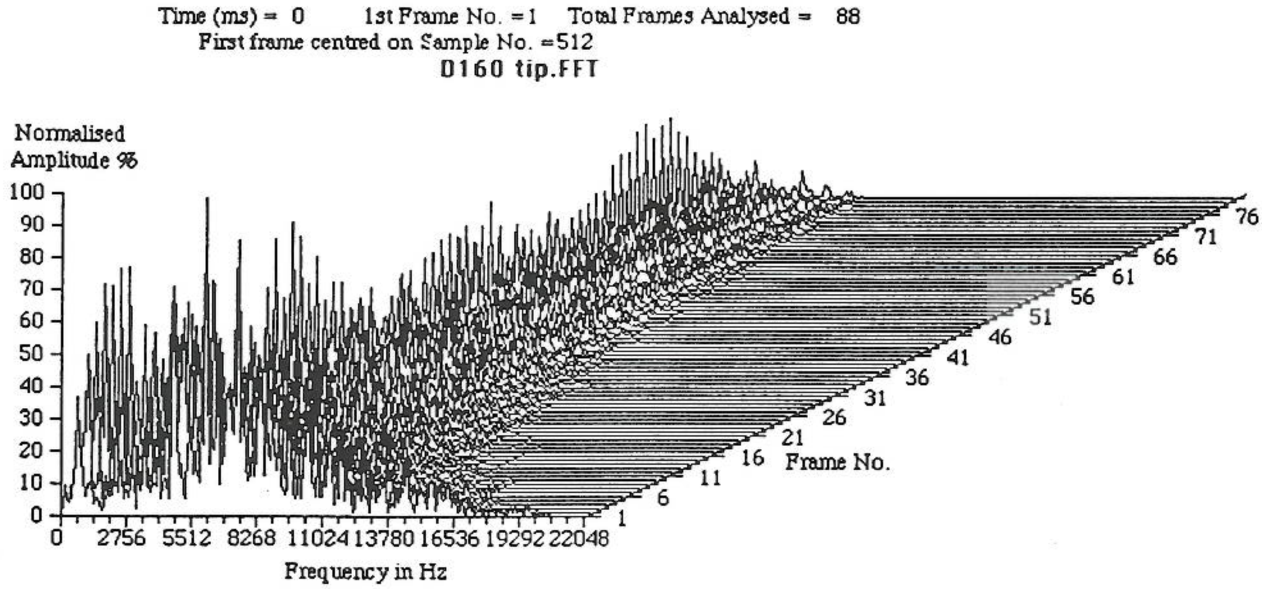
Time (ms) = 0   1st Frame No. = 1   Total Frames Analysed = 88
First frame centred on Sample No. = 512
D160 tip.FFT
Normalised Amplitude %
100
90
80
70
60
50
40
30
20
10
0
0
2756
5512
8268
11024
13780
16536
19292
22048
Frequency in Hz
1
6
11
16
21
26
31
36
41
46
51
56
61
66
71
76
Frame No.

Fig. 1

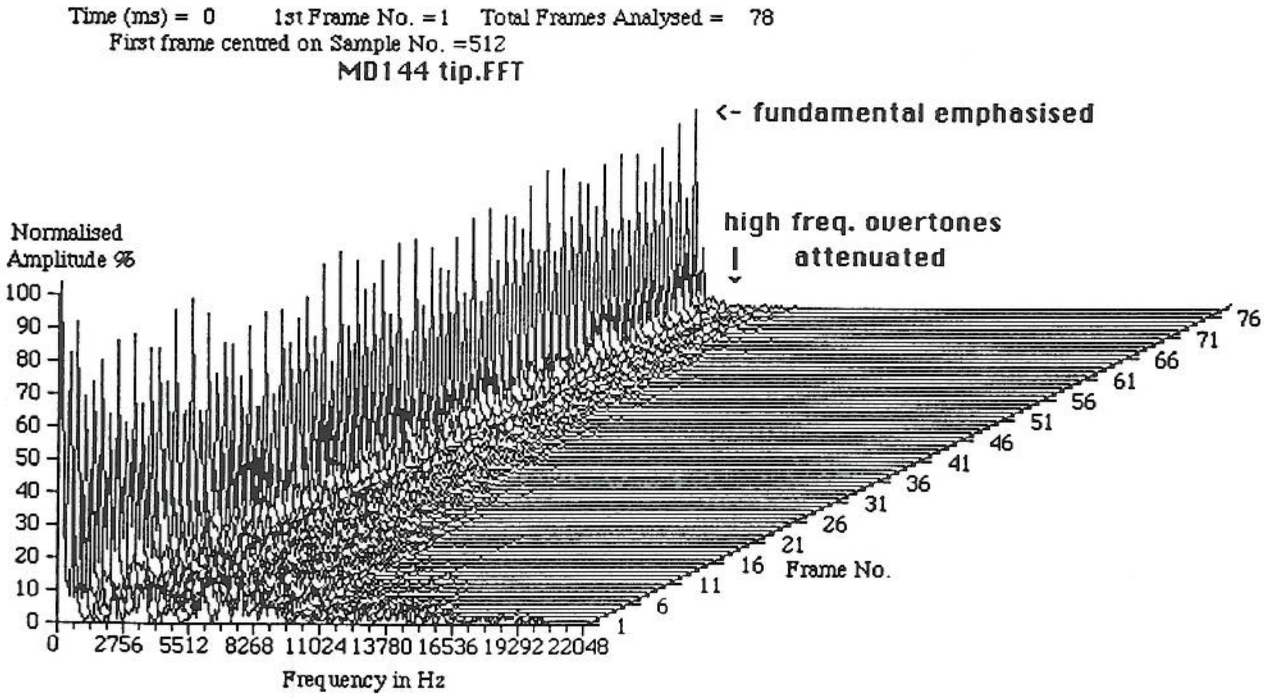
Time (ms) = 0   1st Frame No. = 1   Total Frames Analysed = 78
First frame centred on Sample No. = 512
MD144 tip.FFT
<- fundamental emphasised
high freq. overtones attenuated
Normalised Amplitude %
100
90
80
70
60
50
40
30
20
10
0
0
2756
5512
8268
11024
13780
16536
19292
22048
Frequency in Hz
1
6
11
16
21
26
31
36
41
46
51
56
61
66
71
76
Frame No.

Fig. 2

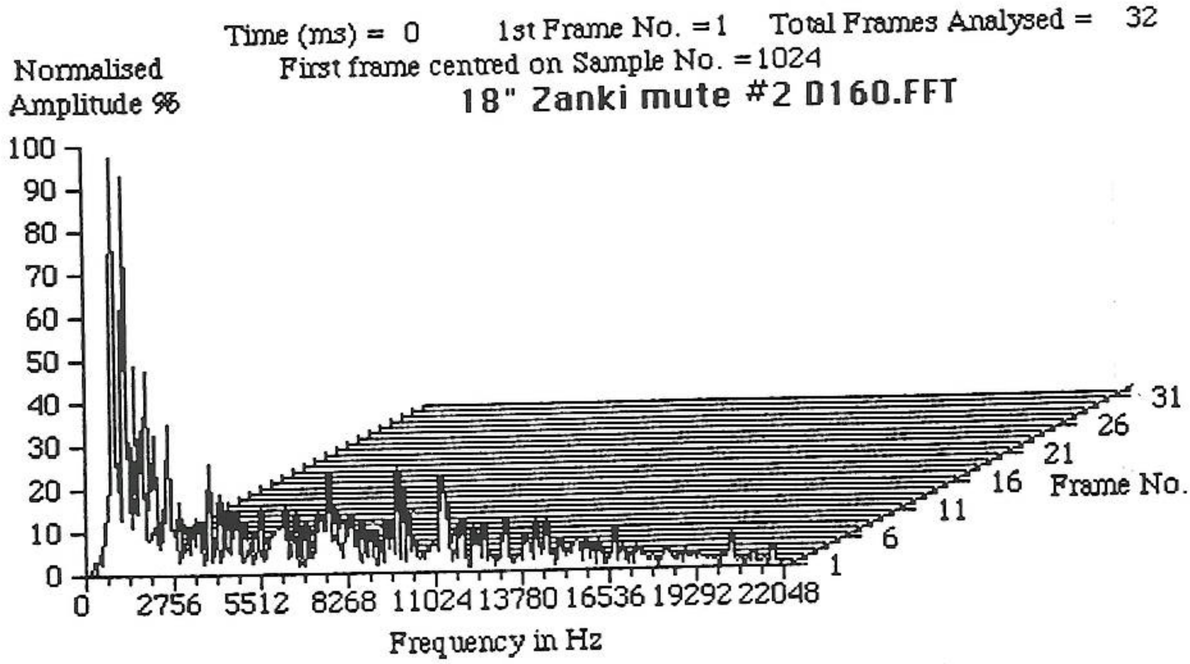

Fig. 3

Real-time control of digital sound processes in performance

- Jim Sosnin, David Hirst and Graeme Gerrard

ABSTRACT

The design of digital signal processing software must address specific problems of dynamic parameter control and module interconnection, if it is intended to be used in music performance. Hardware for such real-time operation is now affordable, and available in a number of commercially distributed development systems. The system used in this project is a Macintosh computer with Digidesign's Sound Accelerator card, which utilises the Motorola 56001 signal processing chip. A number of real-time signal manipulation macros have been programmed in 56001 assembly language. The emphasis is toward more exotic routines that cannot be easily achieved using off-the-shelf black box units; processes such as FM, pitch shifting, gating, ring modulation, granulation, waveshaping, cellular automata, and others have been written. When a number of parallel or serial processes are implemented, the problem of their connectivity arises, also multiplying the problems of real-time parameter control in live performance. A number of patching and control solutions are presented utilising both MAX and our own stand-alone Macintosh interface code. New control interfaces to complement standard MIDI devices are introduced.

AIMS

The project was initiated by the School of Music at the University of Melbourne and by the Music Department of La Trobe University. The aim was to create a software environment for real-time signal processing using the Macintosh and other commercially available hardware, with particular emphasis on the use of this system in live music performance.

Our main interest lies in developing processes that are not usually found in off-the-shelf effects units. Other important features of the software are flexibility, extensibility, and neutrality, so that users should be able to experiment freely by reconfiguring the system. If provided with a modular design, users can build complex configurations from simple units. The desirability of multi-processing leads to a requirement for a method of patching of a number of processes, in parallel or series.

Because the software needs to be robust in live performance, there must be a clear and logical user interface with real-time controls that are practical to use.

DEVELOPMENT SYSTEM

Our development system consists of Digidesign's Pro I/O ADC, their Sound Accelerator DSP56001-based NuBus board (Brooks and Currie 1990), an Apple Macintosh series II computer with MPW 3.1, Motorola DSP56001 Assembler/Linker/Librarian and Simulator (Motorola 1986), Opcode's MAX (Puckette and Zicarelli 1990), and the MaxDSP Development Tools from the University of California at Berkeley (Baudot, Freed, and Gordon 1991). The MaxDSP Tools are also compatible with the MacMix Excelerator board, Digidesign's Audiomedia card, the Ariel DSP56 card, and Polysonic's Reson8 system. MAX versions of our programs should therefore run on any of these systems, but we have developed and tested code only on the Sound Accelerator.

PROGRAMS DEVELOPED

The first stage in the project was to develop a repertoire of signal processing routines in 56001 assembly language. As noted above, the emphasis was on developing more exotic routines that are not generally found in hardware signal processing black boxes which focus more on effects such as reverberation, delay, filtering, flanging, chorusing, and certain types of pitch shifting. The types of routines we have developed so far draw on processes such as Frequency Modulation of an arbitrary audio signal, gating, ring modulation, granulation, non-linear distortion, cellular automata, as well as pitch shifting. These are real-time processes working with real-time controls.

The second stage in the project integrated a number of processing macros into a single package called 'SelectaPro'.

SelectaPro

SelectaPro was written using Digidesign's development software, Symantec's Think C 4.0 and SmethersBarnes' Prototyper. In SelectaPro the 56001 processes audio data while the Macintosh host runs the user interface. All processing control is handled using MIDI data which is input via the Macintosh modem port. MIDI data is converted, in the Macintosh host, to control data that the 56001 can understand, and is passed across the NuBus to the card as often as the MIDI controller determines. SelectaPro uses a custom window (see Figure 1) written using calls to the Macintosh toolbox routines.

When using SelectaPro the user is able to select one of several available processes at run time. While the process is running, various control parameters can be changed dynamically, usually with a combination of MIDI controller, on-screen controls, and characteristics of the sound itself. A typical example configuration would be MIDI modulation wheel controlling an FM modulation amount, an on-screen slider controlling modulation depth, MIDI note number and pitch bend determining modulation frequency, and the envelope of the right channel determining overall output level of the processed left channel audio data. If a MIDI controller is not available, on-screen controls can be used to perform most control functions.

MaxSelectaPro

The MaxDSP object and U.C. Berkeley's MaxDSP development toolkit have been used to develop more complex configurations of processes and to create much more versatile graphical interfaces very quickly.

MAX is an object-oriented graphical programming language that has a library of objects devoted to generating or processing MIDI data, and it includes objects that can manipulate screen graphics such as sliders, pop-up menus, dials, 2D graphs, etc. We have used MAX to provide a MIDI and screen graphics control link between the DSP board and the processing performer. New objects, written in C, can be created and added to the MAX library if a given task cannot be achieved easily or efficiently by combination of existing objects. For example, our research assistant Michael Hewes developed a de-zippering object for smooth transitions between control states; this is now incorporated in MaxSelectaPro. To the end user, the MAX interface has the advantage of being adaptable at run time by simple manipulation of the screen graphics objects, without any changes to the DSP code segments.

MaxSelectaPro performs the same functions as SelectaPro, but with a much more friendly visual interface (see Figure 2). MIDI processing control parameters are displayed on the screen on their way to the DSP board. Screen representations of MIDI controllers, such as a keyboard, can be used if a MIDI device is not available.

MULTIDIMENSIONAL CONTROLS

Test runs of the software and its use in concerts have, to date, involved two or more people in the actual performance. Although the concerts were well received, some performers felt that they had lost control of their sounds during improvised segments. When the acoustic instrument player and the computer process controller are not the one person, their actions need to be coordinated, unless chance effects are being explored. There is an obvious trade-off between the number of instrumental and process control parameters which are potentially variable, and the confidence the instrumentalist has in being able to predict the outcome of playing a given note, for example. This would not be a problem with music which had a detailed score, and which had been more thoroughly rehearsed.

An alternative approach is for the instrumental performer to also control the computer process. However, standard MIDI controllers are not always suitable, especially if the performer has no hands free. New processing control devices can be used in this situation. An ultrasonic 3D locator which generates 12-bit MIDI pitch bend on three channels mapped to x, y, and z coordinates has been successfully tested as a signal processing controller. The small, lightweight transmitter transducer of such a system could be mounted on a (non-MIDI) wind instrument, for example, taking advantage of gestural movements to change up to three processing parameters simultaneously.

A one-dimensional radio frequency version of the ultrasonic locator has also been tested with the system. It has the advantage of a faster response time than the ultrasonic version, but it does not offer the same positional accuracy or linearity. Its virtually instantaneous response allows it to be used when a control parameter is to correspond to velocity, rather than the absolute position, associated with a performer's gesture. Velocity is easily derived from successive position measurements, and may be produced at the analog electronic hardware level, or calculated digitally by the

computer.
Controllers of this general type may be designed around ultrasonic, radio frequency or infrared techniques, and have existed since the days of the Theremin. In recent years there has been a resurgence of interest in their use, and new systems have been created by a various people including Max Mathews, after whom MAX is named.

MULTI-PROCESSING

The final stage of the project is examining the problem of multi-processing, that is, running several processes concurrently to process audio signals in serial and/or parallel configurations. For example, a pitch shifting process could provide the modulator frequency for an FM process.
At the time of writing this is still work-in-process, but our work includes prototypes of several schemes of multi-processor patching. One scheme of dynamic patching utilizes on-screen controls from MAX objects which cause flags to be set. These flags are read during DSP code execution and cause conditional branches within the assembly language routines. This technique allows relatively smooth switching between processes, but has the disadvantage that all the code, for all the 'resident' processes, must be able to fit into DSP program memory, which in our system is currently 4K of 24-bit words; some process collections require code sizes which simply do not fit. Another technique, implemented recently, has only one 'combination' of processes in memory at any time; whenever the user selects a new combination, the appropriate code blocks are loaded into memory, then the processing is restarted. The extra delay is imperceptible, but control parameters exhibit discontinuities on initial use after such reloading.
Planned enhancements include dynamic patching by custom MAX objects, rather than by standard "patcher" objects. It is intended that our custom MAX objects will work directly with audio signal "images" of the processing objects that are "mirrored" on the 56001 board. This is similar in concept to the tilde object approach implemented on the NeXT machine by Miller Puckette (1991).

CONCLUSIONS

In practice it has been found that a versatile patching environment is no less important than the processes themselves. It is often more useful to have a larger number of simple processing routines that can be connected in complicated ways, rather than a small number of complicated processes that can only be connected in simple ways.
SelectaPro and MaxSelectaPro have been used in live performances and have been found to be quite performer-friendly, and have produced interesting results. Sounds from mainly acoustic instruments have been miked and processed by the computer system, to produce new sounds occurring simultaneously with the original sounds. New performance possibilities and some performance problems have been revealed.
Real-time spectral modification has also been considered, and while real-time high fidelity FFT analysis is technically possible using one 56001 DSP board, the combination of FFT analysis, frequency domain modification, and inverse-transform resynthesis, is expected to function with adequate fidelity only on a multi-board system. Other less demanding but very useful processes, such as several types of pitch detection, are being developed for integration into this ever-growing multi-processing application.

ACKNOWLEDGMENTS

This work was made possible by grants from the University of Melbourne and its School of Music, and assistance in kind from La Trobe University's Music Department. We would also like to acknowledge the contribution to this project of our research assistant Michael Hewes.

9. REFERENCES

Baudot, M., A. Freed, and K. Gordon. MaxDSP 56K. CNMAT, Berkeley, California. 1991.
Brooks, E. and R. Currie. Digidesign Developer Documentation. Digidesign Inc., Menlo Park, California. 1990.
Motorola DSP56000 Digital Signal Processor User's Manual. Motorola Inc. Austin, Texas. 1986.
Puckette, M. and D. Zicarelli. MAX Development Package. Opcode Systems Inc., Menlo Park, California. 1990.
Puckette, M. "Combining Event and Signal Processing in the MAX Graphical Programming Environment." Computer Music Journal 15(3): 68-77, Fall 1991.

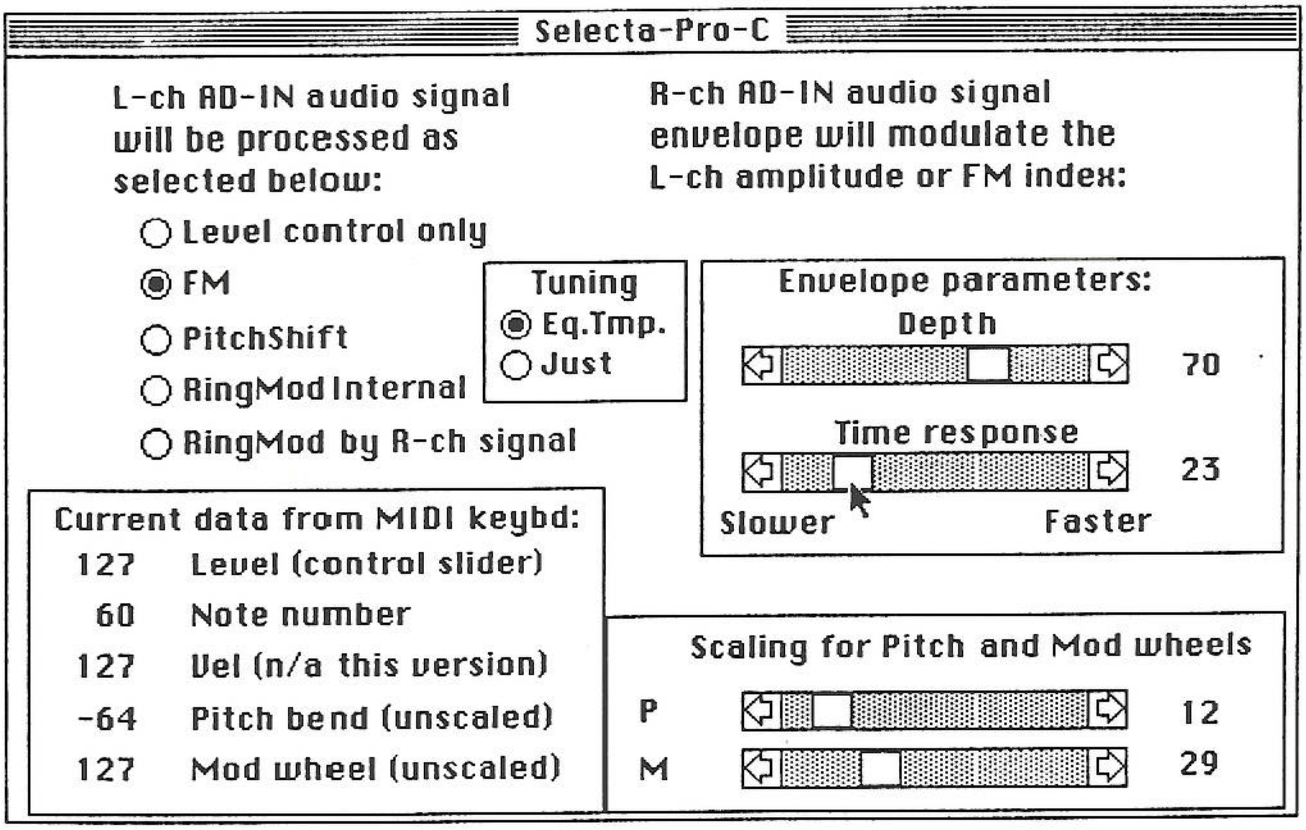

Fig 1. Stand-alone version of SelectaPro

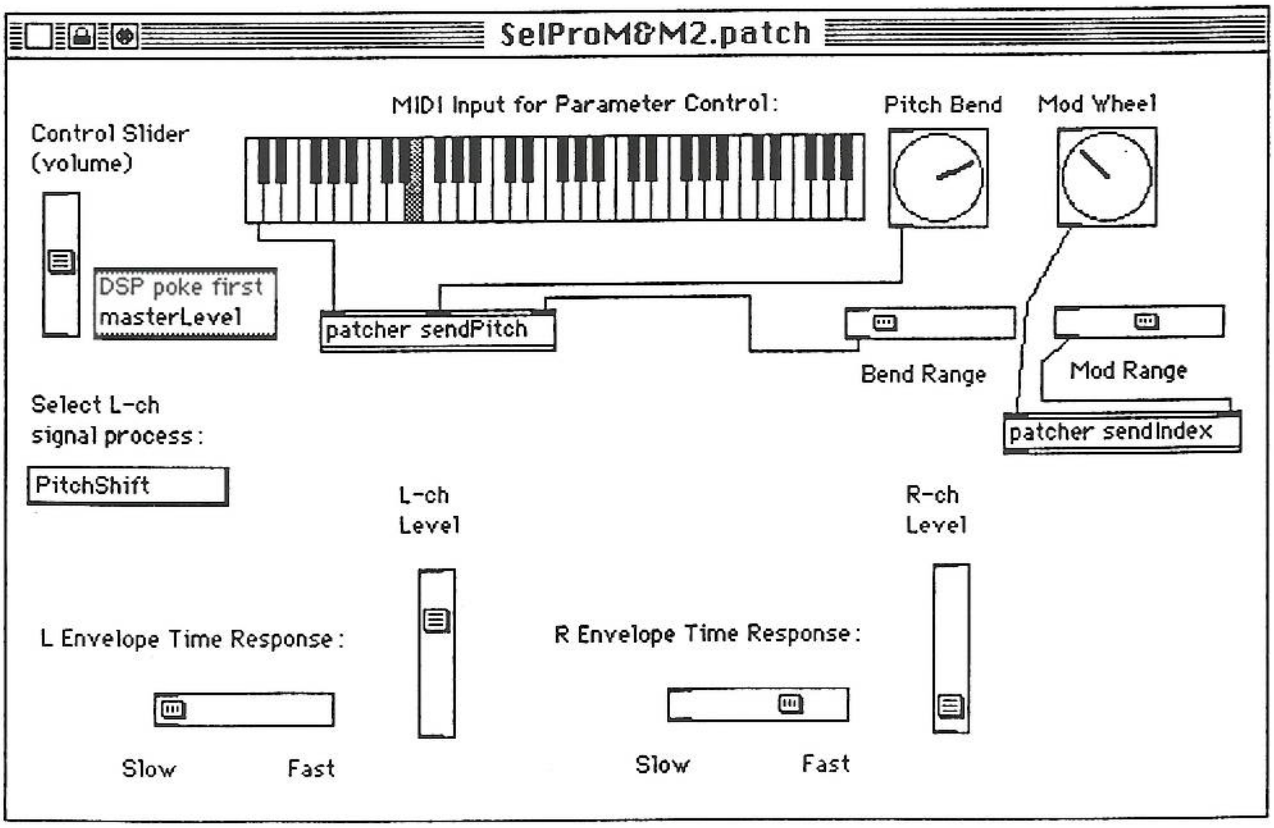

Fig 2. Max version of SelectaPro with better controls

New Software for Spectral Analysis
- Thomas Stainsby

Introduction

AnnaLies2.0 is a program for the spectral analysis of Sound Designer II files, which is run on the Macintosh computer. The analysis procedure used is that of the Short Time Fourier Transform (STFT), which essentially represents the frequency content of a sound as a quantised spectrum. The original version of this program, AnnaLies1.0, was written jointly by David Hirst and Thomas Stainsby in 1991, and was based on research carried out as part of David's Master's thesis [Hirst 1985]. The second version is a fairly comprehensive rewrite, undertaken in 1992 by Chris Scallan and Thomas Stainsby. Significant improvements in the new design include the option of a variety of windowing functions, longer analysis windows, extra display possibilities, a file format which is 60% smaller, and a user interface which is generally more efficient and professional.

The scope of this article does not permit an explanation of the various aspects of STFT analysis, but fortunately there is no shortage of reference material for this topic. A good introductory explanation is that given by Moore (1990: 61-111), while more detailed discussions are given by Allen (1977), Allen and Rabiner (1977), Crochiere (1980), Grey (1975), Griffin and Lim (1988), Hirst (1985), Portnoff (1980, 1981a, 1981b) and Strawn (1987).

The AnnaLies2.0 Program

Operation of the AnnaLies2.0 program follows in the intuitive manner to which Macintosh users have become accustomed. The overall analysis procedure essentially takes a Sound Designer II stereo sound file, performs an STFT analysis of it and stores the result in an FFT (Fast Fourier Transform) file, which can then be called up to have its data displayed in either a 2D or 3D format. The FFT file data can also be read and subsequently manipulated by other programs, such as the TOMQAT program, discussed later in this article.

The following is a quick step by step run through of the operation of the program.

Step 1: The user selects a Sound Designer II stereo sound file for analysis, using a standard Macintosh file selection Dialog. Once the file is selected, the 'Analysis Parameters' Dialog box shown in appears, allowing the user to perform an analysis with the desired parameters. At the top of the Dialog, the sound file name, number of channels, and sampling rate are displayed. The user can select the window length (the number of samples over which the analysis is performed), the hop size (the number of samples between analyses), the type of windowing function used (a weighting function applied to the sample data), and both the first and last samples within the sound file which will be analysed. (See Fig. 1)

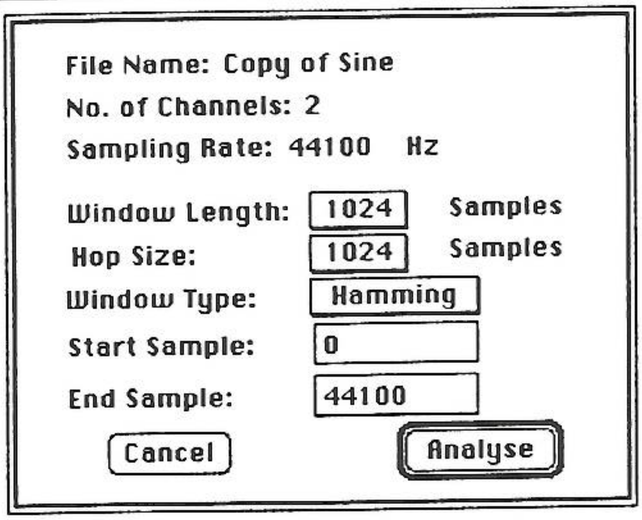

Figure 1: Analysis Parameters Dialog Box

Step 2: Having specified the desired parameters, the user begins the analysis by pressing the 'Analyse' button and the program then prepares to write the analysis data to an FFT file, given the default name of the original sound file followed by a '.FFT' suffix. The user has the option at this point to change the FFT file name if desired. Once the 'save' button in the file name Dialog is hit, the analysis process commences, bringing up the 'Analysis Progress' Dialog. This Dialog contains a print out of the centre sample number of the most recently analysed frame, and a 'terminate' button, which allows the user to terminate the analysis at any point if desired. All the analysis data created prior to the termination will still be saved to file.

Step 3: The user can now select from a range of graphical display options to view the analysis data. These options are selected between in the 'Graph Display Parameters' Dialog, shown in fig. 2. There are five different types of graph, a 2 dimensional Amplitude versus Frequency graph, and four different projection angles (i.e.., backwards or forwards, left handed or right handed), of a 3 dimensional Time versus Amplitude versus Frequency graph. The amplitude scale for all these different graphs can be set to either a linear percentage or decibel value, both referenced to the maximum channel amplitude encountered in the file. The user can also select the first and last frequency channels to be plotted, and determine the analysis frame from which the display will plot.

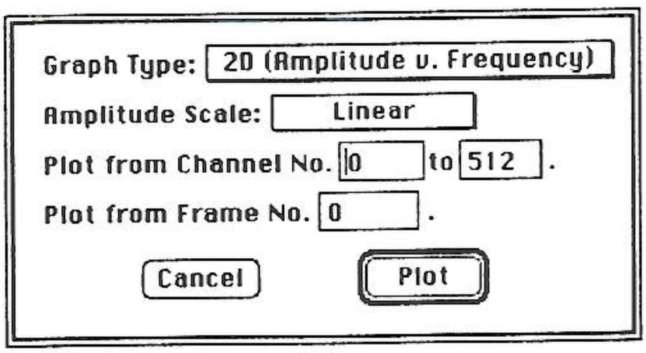

Figure 2: Display Parameters Dialog Box

As an example of the display format, fig. 3 shows a 2D frequency plot of a sine wave made using AnnaLies2.0. Note that the amplitude and frequency values of the point where the cursor is are shown in the top left hand corner of the plot, just under the printout of the analysis parameters used. Fig. 4 shows a 3D plot of the same sine wave shown in fig. 3. This plot was made using the so called 'forward right-handed' projection.

In the article so far, we have looked at the basic operation of AnnaLies2.0, and seen how it can be used to plot the FFT analysis data. In this regard, the program does not break much new ground, although it does provide a useful and practical tool which was hitherto lacking in the Macintosh environment. Where AnnaLies2.0 promises the most, however, is in its application of creating FFT analysis files which can then be used as the starting point for other, more advanced analysis procedures. It is in this area that my chief interest is to be found.

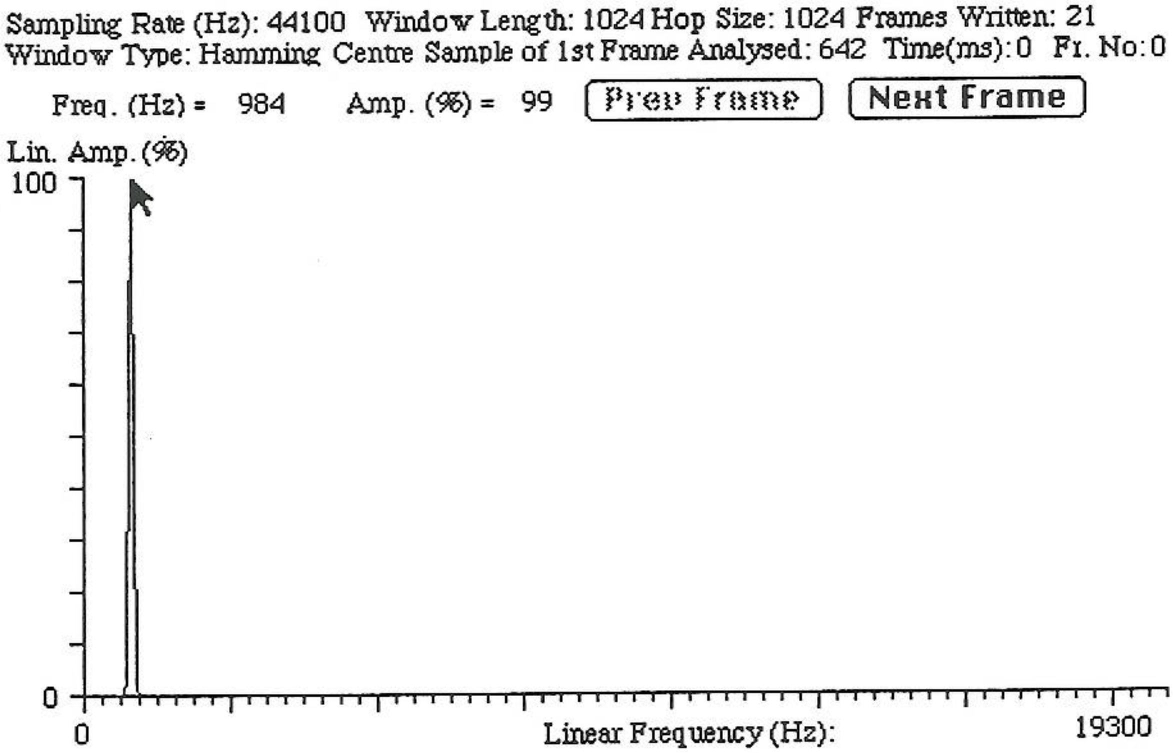

Figure 3: 2D spectral plot of a sine wave

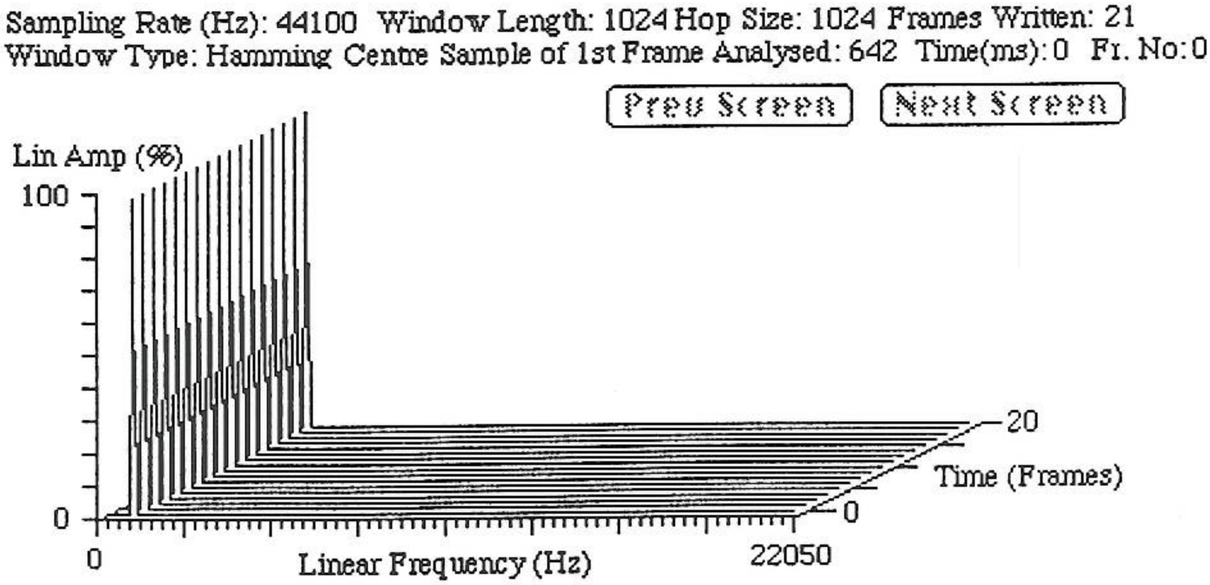

Figure 4: 3D spectral plot of sine wave

AnnaLies2.0 and complex signal analysis and decomposition

My present research lies in the field of complex signal analysis and decomposition, with a particular interest in the way that human listening is capable of distinguishing between simultaneous sounds. This ability has a profound effect on the way we perceive and appreciate most of the music we listen to. All understanding of harmony, counterpoint and orchestration depends on the ability to distinguish between the various instrumental lines or voices, and understand the way they relate to and play off each other. In an effort to begin to emulate this human psychoacoustic ability, we need to create an analysis system which can distinguish between the various sound sources found in a mixed musical signal. The FFT analysis produced by AnnaLies2.0 serves as the ideal starting point for this type of more involved analysis, because it can represent any type of sound in terms of its frequency content. Once in this domain, we can begin decomposing the mixed signal into its constituent voices.

The basic strategy which I have adopted for this type of analysis is to create an additive sinusoidal model of the mixed sound, derived from the AnnaLies2.0 FFT analysis data. To construct this sinusoidal model, we use another original program called TOMQAT, used in [Stainsby 1991]. Having done this, we can then reassemble the component sounds by assigning each partial to a particular voice or instrument. This is similar in concept to the approach taken by Robert Maher (1989, 1990). The representation of a sound as a set of sinusoidal partials is not far removed from the representation of the frequency domain as given by FFT analysis. The analysis procedure used to produce this sinusoidal representation is called MQ Analysis, named after McAuley and Quatieri (1986), who first described the procedure in relation to the analysis of speech signals. A further discussion of this technique is that given by Smith and Serra (1987).

MQ Analysis

The MQ analysis procedure begins by assuming that any peak detected in the frequency spectrum of a sound represents an underlying sinusoidal partial. However, the FFT analysis procedure only specifies the frequency of each analysis channel in values of a multiple of the fundamental analysis frequency, which could typically be in the order of 25 to 100 Hz. This is clearly not of sufficient resolution to accurately specify the frequencies of the partials of musical sounds. A method of finding a more precise frequency value is required. For this purpose, we can use the technique of parabolic interpolation in the amplitude and frequency domains. The three channels immediately surrounding a detected peak can be used to specify a unique parabola. The peak of this parabola has a specific frequency and amplitude, which we can assume to be the same as those of the underlying sinusoidal partial. This process is represented graphically in fig. 5, which is reproduced from [Stainsby 1991].

A series of such peaks in successive frames indicates a continuing sinusoidal track, as shown in fig. 6 [Stainsby 1991]. These tracks can be used to drive an additive synthesis engine, which could then produce a high quality resynthesis of the original sound. The differences between the amplitude and frequency values at successive frames can be adequately compensated for by the use of linear interpolation. To ensure that no audible clicks occur at each end of a partial, corresponding birth and death peaks can be inserted in the frames adjacent to such entries and exits. Such peaks have the same frequency as the matching peak, but have zero amplitude. This allows each partial to fade in or out without producing a click. Note also that this sinusoidal representation of a sound offers great data reduction, as only significant peaks are represented, and then only at each FFT frame analysis point.

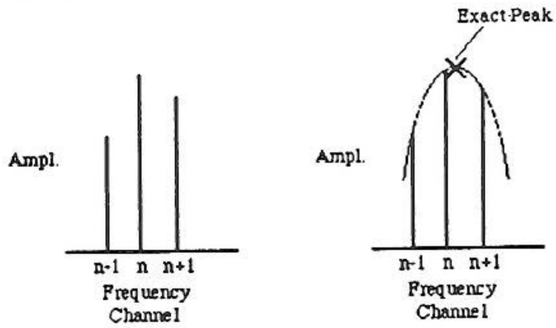

Fig. 5: Parabolic Interpolation of the Frequency Spectrum Using Three Points

Complex Signal decomposition

For the purposes of mixed signal decomposition, we need to be able to assign each partial to a given source sound. Once this has been done, it is possible to resynthesize each sound source independently, and we have thus achieved a decomposition of a complex musical signal into its component sound sources.

It is immediately apparent then, that the relative success of a decomposition technique depends on the criteria and method used for assigning a given partial to one voice or another. In my work to date, the primary criterion has been the ratios between frequencies of individual partials. Those partials whose frequencies were in a harmonic ratio to a given fundamental were assumed to belong to that voice.

Fig. 7 shows an analysis example taken from my research work last year [Stainsby 1991]. This shows a 2 dimensional plot of a flute and 'cello playing a diad with the interval of a major sixth. All the partials from both instruments can be seen here. Using the TOMQAT program, the partials which are judged to be harmonic multiples of the 'cello fundamental will be extracted and assigned to the lower voice, to be resynthesized as the 'cello. The fundamental frequency of the lower voice is determined to be the same as the frequency of the first sufficiently strong partial detected within a predefined lower voice fundamental frequency range. The partials from the mixed file which are not assigned to the lower voice can then be assigned to the upper voice. In this case, they would represent the flute.

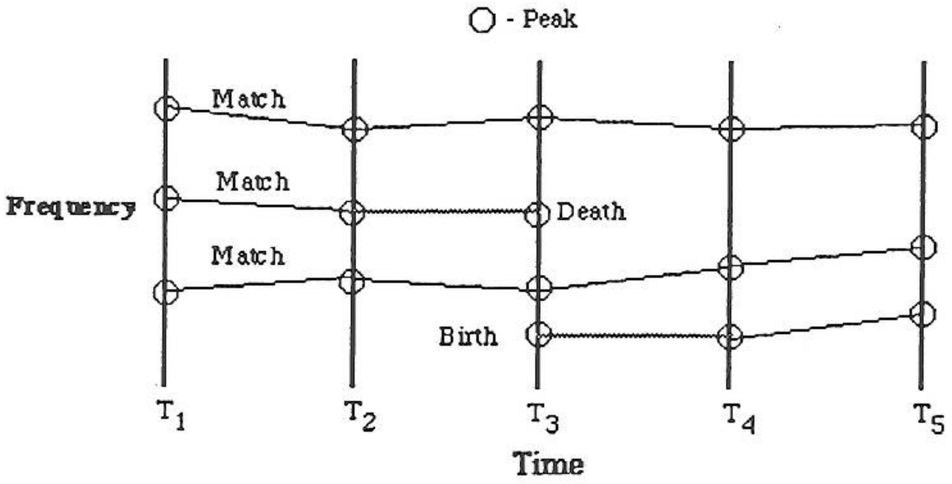

Fig. 6: Peak matching procedure in MQ analysis, showing matches, births and deaths.

Conclusion

In this paper, a brief description of a new program for musical signal analysis on the Macintosh computer was given, along with a description of one particular application for the program beyond the immediate use of simply producing plots of a sound's frequency spectrum. By using the TOMQAT program, also written by the author, in conjunction with the AnnaLies2.0 program, a preliminary method of mixed signal decomposition was demonstrated. Future work in this area requires a more robust method of partial assignment, to allow for the separation of sounds which contain non-harmonically related partials. Research into this area continues to be undertaken by the author.

BIBLIOGRAPHY

Allen, Jont B., 1977, "Short term spectral analysis, synthesis, and modification by discrete Fourier transform", IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. ASSP-25, no. 3, pp. 235-238.

Allen, Jont B., and Rabiner, Lawrence R., 1977,"A unified approach to short-time Fourier analysis and synthesis", Proceedings of the IEEE, vol. 65, pp. 1558-1564.

Crochiere, R. E., 1980, "A weighted overlap-add method for short-time Fourier analysis/synthesis", IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. ASSP-28, no. 1, pp. 99-102.

Grey, J. M., 1975, An Exploration of Musical Timbre, Ph.D. dissertation, Stanford University, Stanford, California, (also Dept. of Music rep. STAN-M-2).

Griffin, D. W., and Lim, J. S., 1988, "Signal estimation from short-time Fourier transform", IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. ASSP-32, no. 2, pp. 236-242.

Hirst, David J. G., 1985, Digital Sound Analysis and Synthesis Using the Short-Time Fourier Transform, M.A. Thesis, La Trobe University Music Department, Melbourne, Victoria.

McAulay, R. J., and Quatieri, T. F., 1986, "Speech analysis/synthesis based on a sinusoidal representation", IEEE Transactions on Acoustics, Speech and Signal Processing, vol. ASSP-22, no. 5, pp. 330-338.

Maher, Robert C., 1989, An Approach for the Separation of Voices in Composite Musical Signals, Ph.D. dissertation, University of Illinois, Urbana.

Maher, Robert C., 1990, "Evaluation of a Method for Separating Digitized Duet Signals", Journal of the Audio Engineering Society, vol.38 no.12, pp. 956-979.

Moore, F. Richard, 1990, Elements of Computer Music, Prentice Hall, Englewood Cliffs, New Jersey.

Portnoff, Michael R., 1976, "Implementation of the phase vocoder using the short-time Fourier transform", IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. ASSP-24, no. 3, pp 243-248.

Portnoff, Michael R., 1980, "Time-frequency representation of digital signals and systems based on short-time Fourier analysis", IEEE Transactions on Acoustics, Speech and Signal Processing, vol. ASSP-28, no. 1, pp. 55-69.

Portnoff, Michael R., 1981a, "Short-time Fourier analysis of sampled speech", IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. ASSP-29, no. 3, pp. 364-373.

Portnoff, Michael R., 1981b, "Time scale modification of speech based on short-time Fourier analysis", IEEE Transactions on Acoustics, Speech, and signal Processing, vol. ASSP-29, no. 3, pp. 374-390.

Smith, Julius O., and Serra, Xavier, 1987, "PARSHL: An analysis/synthesis program for non-harmonic sounds based on a sinusoidal representation", Proceedings of the 1987 International Computer Music Conference, Computer Music Association, San Francisco, California, pp. 290-297.

Stainsby, Thomas H., 1991, The Isolation of Musical Streams in Polyphonic Audio Signals, B.A. Honours thesis, Music Department, La Trobe University, Bundoora, Victoria.

Strawn, John, 1987, "Analysis and synthesis of musical transitions using the short-time Fourier transform", Journal of the Audio engineering Society, vol. 35, no. 1/2, pp. 1236-1249.

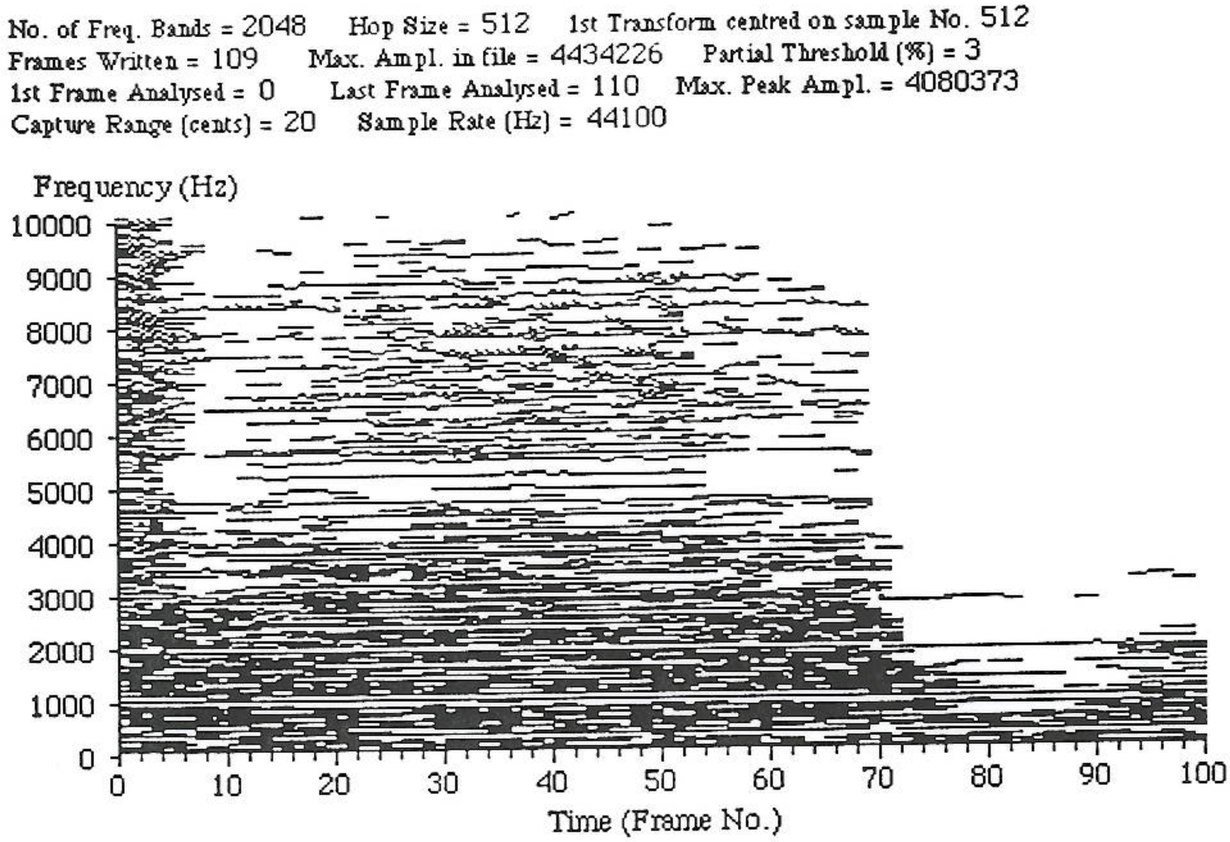

Fig. 7: 2D MQ plot of 'cello and flute diad (from TOMQAT)

Sound Colour: the listener hears it

- Linda Ceff

A consideration of the notion of *archetype* has implications for our understanding of how we perceive sound that are bypassed in studies that treat sound only as separate from a listener. Examining definitions and uses of this concept, and related issues of timbre, can provide a background for exploration of ways that we may define the functioning of our musical awareness. This includes reference to existential and gestalt concepts, in consideration and provocation of the very basis of how it is that we experience.

In this article, there is minimal content of relatively traditional material, where sound is treated as an object separate from the experience of hearing, due to the high quality and quantity of this type of research elsewhere in this edition of *Chroma*. Many studies have been made focusing on the sound spectrum, and sophisticated digital processing procedures have led to some interesting and exciting compositions. As more work is done in this area, our understanding of acoustics, and of how sound functions, is sure to be expanded. In particular, I have used graphs from Digidesign's *SoundDesigner II* program, *Annalies2.0* by Thomas Stainsby and Chris Scallan, and *Spectrum* by Graeme Gerrard. [1] Presentation of various graphic representations of sound in seminar situations, along with aural samples, provides a broader perspective on sound, extending that which is possible with traditional manuscript and formal analyses. Although the positivist approach to analysis and its achievements are seminal to the use of computers for composition and as tools assisting the experience of music in the future, its greatest benefit will be when this attitude and materials are integrated with parallel advances in other disciplines such as psychology, nuclear physics, and religious studies. An interesting development in my research has been that the few areas of focus that I initially regarded as basically separate approaches to the phenomenon of sound colour have become increasingly more integrated, with the recognition of various common concepts in different fields, which although sometimes couched in different language, are coming together within a domain of colour perception and definition.

Colour describes the archetypal nature of the total combination of frequencies - *sound* colour describes them for the period of time that it takes to perceive such a gestalt, in order to define the colour. For humans hearing sound, durations of less than between about five to twenty milliseconds are not generally recognizable as variable timbres. Perceptual boundaries are integral to definitions of these factors. The experience of music occurs throughout a unique period of time, and the aspect of colour describes attunement to experience and definition of the moment. Colour is multi-dimensional in that it can refer to visual, aural, or other ways of experience, with specific kinds of vibration recognizable as patterns of archetypal qualities. The generation of specific frequency bands in various intensities can create a particular visual colour, another range may generate a sound, another may kill bacteria, another may generate a visual representation of a form (ultrasound), another may prolong the onset of decay in food products, yet another may break glass: Although each of these examples is from a different medium, they all share colour as a definable aspect of quality in terms of description of vibration.

The term *archetype* derives from ancient Greek; *Arkho* meaning begin or rule. [2] During this century, Carl Jung used the term to describe those mysterious inner forces that we acknowledge but cannot explain. He writes:

These are indefinite, that is to say they can be known and determined only approximately. Although associated with causal processes, or 'carried' by them, they continually go beyond their frame of reference, ... the archetypes are not found exclusively in the psychic sphere, but can occur just as much in circumstances that are not psychic (equivalence of an outward physical process with a psychic one). ... The archetype represents psychic probability, *portraying ordinary instinctual events in the form of types.* [3]

What this means for sound colour perception, is that when we hear a sound, such as a piano chord, for example, the experience and definition of the colour is unique to the moment of hearing, during the manifestation of the archetype. The listener recognizes patterns within the sound, that may be observable in graphic analysis, but the pattern does not cause or define the sound colour. The archetypal pattern is an abstraction that never manifests in exactly the same way. Each listener experiences gestalten in a forever changing way, that is never totally predictable, although patterns and associations may be found that enable a probable prediction.

Studies and practices of gestalt perception have been made by increasing numbers of people in the last few decades. Gestalt thinking is more a philosophical way of being, akin to Zen, rather than a specific theory or formula that can exist outside of any particular person. Gestalt is a word for an orientation, a description of the process involved in individual human awareness and functioning. In contrast with studies of specific events or variables, that tend to use large numbers for statistical validity, gestalt dares to focus on the moment-to-moment functioning of an individual in all its detail and complexity. Stockhausen referred to Gestalten in 1952:

... a series offering the most comprehensive and meaningful potential for change and renewal - without foreseeable limit. One never hears the same thing twice. Still, one is always aware of an unchanging and underlying constancy of flow permeating the whole. A latent power, that holds together related proportions: a structure. Not the same Gestalten *in a changing light. Rather: different* Gestalten *in the same light, that penetrates everything.* [4]

The relationship of twentieth-century physics with the idea of the collective unconscious and manifestation of archetypes has been explored extensively by Jung:

If the latest conclusions of science are coming nearer and nearer to a unitary idea of being, characterized by space and time on the one hand and by causality and synchronicity on the other ... it seems to show that there is some possibility of getting rid of the incommensurability between the observed and the observer. The result, in that case, would be a unity of being which would have to be expressed in terms of a new conceptual language. [5]

In this last point, Jung is making reference to the limitations of language that have evolved as an expression of a way of thinking that is outdated and often inappropriate and/or misleading, in relation to the new concepts that are trying to be expressed. The main relevance of the above quotation to sound perception is the simultaneous existence of the observed and the observer, which in music is the sound and the hearer, and the connection of this apparent polarity in the domain of archetype. The sound colour that the listener hears is experienced as a manifestation of an archetype.

Generally, sound archetypes manifest as particular instrumental sounds. We have traditional archetypal groups in the form of instrumental families. Any one particular hearing of a sound is never the same as another, although it may relate to other sounds that have been heard, and be seen in a graphic analysis to have similar patterns. A significant feature of electronic music is the capacity to expand the familiar instrumental archetypes, and conventional ways of achieving timbral definition. This introduces the notion of musical landscape. Context is an important factor in how we perceive sounds, that has been explored by the *musique concrete* school, and more recently by composers such as Trevor Wishart.

Timbre is an elusive phenomenon, which is defined even by Jean-Claude Risset, initially, in terms of what it is not: 'Timbre, this attribute of the tone that can distinguish it from other tones of the same pitch and loudness, is also called tone color or tone quality.' [6] It is a fairly effective means, in this case, to understand one sense by removing factors of other senses, in consideration of the traditional European repertoire, and the unequal role of aspects such as melody, harmony, rhythm and dynamics. We understand and feel a difference between two different instruments that play the same (fundamental) note, but this phenomena has not been given the same attention as (fundamental) pitch and rhythmic configurations. *Timbre* is also defined in the Oxford Dictionary in this negative way, as the 'characteristic quality of a musical sound or a voice apart from its pitch and intensity'. [7] The same reference also defines the *quality* of a voice or sound as the *timbre*. Such circular and non-affirming definitions do little to describe this phenomena, as they attempt to apply the same criteria as has been used for traditional harmonic and rhythmic systems, which are not

adequate enough in themselves for applying to unique individual experience.
The archetypal nature inherent to timbre is what connects past and present uses of the term. Its earliest known application was by French scholars in medieval monophony, 'to characterize standard melodic themes, phrases or neumatic formulae'. These phrases occur in different musical compositions with different words. The term has been applied to various stages of melodic maturity; a catalogue of 183 *timbres adamiens* (c.1170) included the hypothesis that these structurally simple tunes (recurring melodic phrases) derive from a collection of medieval tunes *les timbres populaires*. More recently (1925), the term *timbres gregoriens* has been used to describe 'stereotype melodic and rhythmic groups of notes or stock neumatic formulae that often occur in certain classes of plainchant melodies. Discussions have tended to focus on the adaptation of secular musical sources for liturgical compositions. In the late eighteenth-century, a *timbre* could be derived from pre-existing *opera-comique* songs, vaudeville tunes, parody songs, sixteenth- and seventeenth-century chansons, and particularly medieval monophony; the name of the timbre was taken from the refrain or first couplet of the original poem to identify the tune when it was set with new words. [8] Although these uses apply to melodic configurations, the idea of prototypical units is the significant and connecting feature of the various definitions.
Historical uses of the term *colour* also emphasize the medieval Latin use, to signify embellishment and particularly repetition. It referred to melodic repetition, as distinct from rhythmic repetition (talea), in a similar way to the use of *timbre*. Theorists of the fourteenth- and fifteenth-centuries refer to colour as the presence of chromaticism, in the sense of dissonance creating colour in relation to consonance. The whole tone was divided into unequal parts 'with which we perfect the imperfect [intervals] and colour them.' [9] *Coloratura* (Italian) and *Koloratur* (German) refer to ornamentation that has come to be associated with seventeenth- and eighteenth-century vocal music. *Coloration* was the writing out in detail of such ornamented passages that were initially improvised. [10]
Spectrum is defined as ' the coloured band into which a beam of light is decomposed', and *spectral analysis* as 'the ascertainment by the spectroscope of the elements composing a body'. The derivation from *spectre* emphasizes the inherent mysteriousness: 'ghost-like, unreal in appearance or sound, of the spectrum'. [11] There is intangibility innate within the definition of this phenomena. *Colour* is defined as 'any one, or any mixture, of the constituents into which light decomposes as in rainbow or spectrum'. [12]
Chromatic is defined as 'of colour, in colours', and specifically for music as having 'notes not included in the diatonic scale'. The origins are from the Greek *Khroma*, meaning 'colour'. [13] *Chroma* has also been used to describe the remaining higher partials of a sound after the fundamental has been removed. [14] Also called 'Triple level pitch quality' or the 'Schouten effect' [15] , a sound that has a harmonic series present, but no fundamental, will be perceived as having a fundamental pitch anyway. The chroma contributes significantly to the colour of a sound, as well as reinforcing the basic pitch.
Sound colour refers to the aspect of sound that is generally understood as a vertical dimension. Usual graphic representation of music, such as lines with clefs (fundamental pitch/time) and amplitude/time graphs, show the rate of time passing towards the right in a horizontal dimension. Timbral information is included in classical Western notation to the extent that harmony, and performance and scoring directions describe the sound. The presence of patterns, or lack of patterning in the distribution of partials can be recognized as timbral features.
Individuals vary greatly at timbral recognition, and experience with certain instruments and sounds is important for timbral definition. Single spectra are useful for characterizing the timbre of an arbitrary sound within the limits of the graphic dimensions. A sound spectograph displays the temporal evolution of the spectrum, which has provided new visual representations of various kinds of music. Our hearing process seems to have two significant, and to some extent definable ways of assessing a sound: the nature of the excitation or attack, and the general structural properties of its would-be stationary sound. The listener makes internal reference to an archetype that has been identified in earlier experience. The existence of archetypal qualities, that can be experienced as the colour, seems integral to the phenomenon of musical experience; although our way of identifying these archetypal qualities is culture-bound, their presence is not.
Standard musical references, such as *The New Grove* reserve the term *timbre* for describing the steady state of a sound, and associate it with the physical quantity measurable in the partials. [16] Helmholtz and others studied almost exclusively the most stationary part of the sound. This established the notion that timbre is determined according to the distribution and relative intensity of the partials in the spectrum, in an average calculated for some duration. Later, instrumental recognition was found to be impaired significantly when the initial segments of notes were removed. The attack came to be emphasized as an important clue for recognizing instruments.

These revelations suggest an entirely new categorization of musical instruments according to purely acoustic criteria; throw light on previously unrecognized affinities between instruments of different families; and - most important at the time [1950s] - suggest the possibility of creating new timbres by transplantation of attacks to different resonances. [17]

Details of the decay are particularly important for the timbre of instruments with low damping, such as bells, gongs and low piano tones. The environmental context is more significant for these types of sounds; the acoustic nature of the room has more time to be present in what is heard. The terms *timbre* and *sonance* were used by Seashore in 1938 in a similar way to Schaeffer's 1966 concepts *matiere* (sound material, as defined by a spectral cross section) and *forme* (sound envelope, changing over time). [18] A law of variation, rather than an invariant, can sometimes be useful in characterizing instruments or instrument families. For example, the piano from the treble to the bass has the attack becoming less abrupt and the spectrum richer. [19]

We all constantly exercise our capacity to listen in a direct and subjective way. Associations and patterns are recognised that can give a coherence and consistency to types of experience, and the nature of the vibrational manifestation. The phenomenon of sound colour invokes forces that are inherently mysterious; and attempts to explain and define these issues necessarily involve questioning and expanding the uniquely changing nature and quality of every moment of experience.

Endnotes:

[1] These programs run on a Macintosh computer. See 'New Software for Spectral Analysis', by Thomas Stainsby, in this edition of *Chroma*.
[2] *Oxford Dictionary*, p.35.
[3] Jung, *Synchronicity*, p.99.
[4] Bearings 1952/53, in Maconie, p.35.
[5] Jung, *Synchronicity*, p.96.
[6] Risset, p.14.
[7] *Oxford Dictionary*, p.878.
[8] *The New Grove*, Emerson, p.823.
[9] *The New Grove*, Sanders and Lindley, p.584.
[10] *The New Grove*, Donington, p.584.
[11] *Oxford Dictionary*, p.799.
[12] *Oxford Dictionary*, p.150.
[13] *Oxford Dictionary*, p. 133.
[14] Meyer-Eppler, 'Stastistic and Psychologic Problems of Sound', *Die Reihe* I, in Maconie, p.75.
[15] Maconie, p.75.
[16] *The New Grove*, Emerson, p.823.
[17] Maconie, p.42.
[18] Seashore, *Psychology of Music*.
[19] Schaeffer,1966, in Risset, p.30.

Bibliography

Burge, David L. *Perfect Pitch: Color Hearing for Expanded Musical Awareness.* Presque Isle, U.S.A.: American Educational Music Publications, 1985.

Cogan, Robert. *New Images of Musical Sound.* Cambridge, Massachusetts and London, England: Harvard University Press, 1984.

Erikson, Robert. *Sound Structure in Music.* California: University of California Press, 1975.

Fowler, F. G., and H. W. Fowler, compilers. *The Pocket Oxford Dictionary of Current English,* 4th edition, revised by H. G. Le Mesurier and E. Macintosh. London: Oxford University Press, 1961.

Jung, Carl G. *The Archetypes and the Collective Unconscious,* 2nd ed. trans. R.F.C. Hull, Vol 9, I, in the Bollingen series of Collected Works. Princeton: Princeton University Press, 1969.

Jung, Carl G. *Synchronicity: An Acausal Connecting Principle,* trans. R.F.C. Hull, extracted from *The Structure and Dynamics of the Psyche,* in the Bollingen series of Collected Works. Princeton: Princeton University Press, 1973.

Kandinsky, Wassily. *Concerning the Spiritual in Art.* New York: Dover, 1977.

Krumhansl, Carol L. 'Tonal and Harmonic Hierarchies', in *Harmony and Tonality:* Papers given at a seminar organized by the Music Acoustics Committee of the Royal Swedish Academy of Music. Stockholm: Royal Swedish Academy of Music, 1987.

Leppert, Richard and Susan McClary, ed. *Music and Society: the Politics of Composition, Performance and Reception.* Cambridge: Cambridge University Press, 1987.

Lewis, Robert C. *The Sacred Word and its Creative Overtones; Relating Religion and Science through Music.* California: The Rosacrucian Fellowship, 1986.

Manning, Peter. *Electronic and Computer Music.* Oxford: Clarendon Press, 1987.

Mathews, Max V., with Joan E. Miller, F. R. Moore, J. R. Pierce, and J.C.Risset. *The Technology of Computer Music.* Cambridge, Massachusetts and London, England: The Massachusetts institute of Technology Press, 1981. 1st ed. 1969.

Portmann, Adolf, and Rudolf Ritsema, eds. *Color Symbolism: Six Excerpts from the Eranos Yearbook 1972.* Dallas, Texas: Spring Publications, 1988.

Risset, Jean-Claude. *Musical Acoustics.* U.E.R. de Luminy, Universite d'Aix Marseille,13288 Marseille CEDEX 2, France. 1972, revised 1977.

Sadie, Stanley, ed. *The New Grove Dictionary of Music and Musicians* London: Macmillan, 1980.

Seashore, Carl E. *Psychology of Music.* New York: Dover, 1937.

Silva, Matteo. 'The Magic of Overtone', *The Mirror,* (International Newspaper of the Dzog-chen [Tibetan Buddhist] community founded by Namkhai Norbu Rinpoche). Italy: Associatione Culturale Communita Dzog-chen, April 1992.

Steiner, Rudolph. *The Inner Nature of Music and the Experience of Tone,* selected from *Das Wesen des Musikalischen und das Tonerlebnis im Menschen,* translated by Maria St. Goar, edited by Alice Wulsin. Hudson, New York: Anthroposophic Press, 1983.

Sundberg, Johan. 'Harmony and Harmonic Spectra', in *Harmony and Tonality:* Papers given at a seminar organized by the Music Acoustics Committee of the Royal Swedish Academy of Music. Stockholm: Royal Swedish Academy of Music, 1987.

Tenney, James. *META HODOS: A Phenomenology of 20th-Century Musical Materials and an Approach to the Study of form* and *META Meta-Hodos.* Hannover, N.H.: Frog Peak Music, 1988.

Tipler, Paul A. *Physics.* New York: Worth Publishers, 1978.

Winckel, Fritz. *Music, Sound and Sensation: A Modern Exposition,* translated by Thomas Binckley. New York: Dover,1967.

Wishart, Trevor. *On Sonic Art.* 83 Heslington rd, York Y01 5AX, U. K. : private publication, 1985.

Wishart, Trevor. 'What is musicmontage?', in *Electronic Music for Schools,* ed. Richard Orton. Cambridge: Cambridge University Press, 1981.

Zukav, Gary. *The Dancing Wu Li Masters: An Overview of the New Physics.* New York: Bantam, 1989.