GCOMP: GRAPHIC CONTROL OF MIXING AND PROCESSING

Colin Banger and Bruce Pennycook

Queen's University
Computer Music Facility
Kingston, Canada

November, 1981

GOOMP : GRAPHIC CONTROL OF MIXING AND PROCESSING

=====

USER GUIDE FOR DCIEM

February, 1982.

GOOMP, written by Colin Banger under the M.Sc. thesis supervision of Bruce Pennycook, is an interactive graphics system for the declaration, modification and display of time varying functions. These functions are then submitted to a music synthesis device or language to control the most significant parameters for the mixing of previously generated sound files.

The synthesis language must conform to the following specifications:

1. read external function files of arbitrary table.
(the table length defaults to 256 integer words but can be modified in the output module of gcomp)
2. read external sound files
(cmusic requires the soundin unit generator)
3. read time ordered lists of 'note' data for each input instrument.

The following local modifications are necessary:

1. installation dependant communications with the Norpak IGP must be completed.
2. GOOMP output must be modified to:
 - conform with the external function read format of cmusic
 - conform with the NOTE format of cmusic for each mix 'instrument'
 - conform with DCIEM soundin formats

GOOMP has been proven under test conditions at QUOMF. These tests were all completed using MUSIC 11 formats. The interface to cmusic, the current synthesis language at QUOMF, has not been made as yet.

DISCLAIMER

=====

The author of GOOMP and personnel of QUOMF agrees to provide GOOMP revisions up to and including March 31, 1982. Any further updates will be provided only if possible and practical.

GCOMP: A Graphics Based Sound File Mixing System

Colin Banger and Bruce Pennycook

(Abstract)

The general practice of sectional generation of computer compositions combined with an increasing trend toward digital recording and processing of live music has led to a need for a comprehensive sound file mixing system. GCOMP enables the composer to graphically construct time varying control functions for several processes normally associated with analogue mixing sessions, align and edit these functions, and maintain a wide range of interdependent controls over the output data. The system automatically generates 'score data' for insertion in MUSIC V type synthesis languages which have a sound file reading capacity. Operations include: amplitude envelopes, distance/location/reverberation control functions, four band equalisation and/or filtering, entry timings, and automatic multi-channel timing synchronisation. A special data set stores pointers to segments of sound files. These timing pointers can be generated in either GCOMP or a graphics sound file editor thus eliminating the need for multiple copies of audio sample data. All display routines have been isolated from the command structures to facilitate portability. Much of this work constitutes C. Banger's MSc. Thesis in Computing Science and has been supported in part through a research contract with the Canadian Defense and Civil Institute for Environmental Medicine (CFB, Downsview).

GCOMP: Graphic Control Of Mixing and Processing

abstract

1. current mixing tools
2. Queen's University Computer Music Facility
3. Design Strategies
 - 3.1. high level interaction
 - 3.2. studio like operations
 - 3.3. independence from signal processing software
 - 3.4. independence from processing algorithms
 - 3.5. generality and portability
4. GCOMP design
 - 4.1. user interface (i/o,graphics)
 - 4.2. Modes
6. Evaluation
7. Conclusion
8. References

1. Current Mixing Tools

In an effort to decrease computation loads, most studios at this time have some software tools for mixing and scaling digital sound files. These programs vary from simple entry-delay-time plus amplitude scalar format to more complex packages that permit multi-channel distribution, fader simulation and reverberation. Their main purpose is to free what is essentially a rather simple add + scale process from the general purpose music software (which tends to be too large and slow). Users often find these programs to be somewhat frustrating as fine tuning mix levels by numerical value is often arbitrary or misleading.

Real-time digital mixing systems are not yet common although the commercial recording industry promises several entries in this market in the near future. Currently some digital synthesis systems perform sound file mixing but disc bandwidths which limit the maximum throughput continue to be a bottleneck for multi-channel applications.

A system in use by the BBC is described by G.W. McNally [1979]. This system used an experimental multitrack digital tape recorder and a specially designed programmable mixing desk which included 12-->2 fader assignment, equalization and companding.

It is my understanding that the system being constructed by James A. Moorer and John Snell [unp.] for Lucasfilms will contain new and important breakthroughs in the technology of real time sound mixing.

2. Queen's University Computer Music Facility

Diagram #1 illustrates the basic configuration of QUCMF.

QUCMF is a software synthesis system with a primary focus on digital musique concrete. Thus it was decided that graphics based editing and mixing tools would be essential. Some of these tools are now complete, GCOMP being the first.

Sound synthesis and processing is done on a VAX 11/750 using UCSD cmusic. This has recently replaced MUSIC-11 from MIT which ran on a PDP 11/45. From my brief experience with cmusic it is clear that for our purposes, this software is capable of providing extremely flexible tools for sound file handling, processing and algorithm modelling.

The long term goals at this studio include real-time implementation of the techniques developed in the software environment. (The author is currently involved with two other projects, both of which include real-time array processing devices. It is likely that some of the mixing/processing methods will be developed through this work).

QUEENS'S UNIVERSITY
COMPUTER MUSIC FACILITY

Diagram #1.

3. Design Strategies

Five basic design strategies were adopted which satisfy both the musical/compositional demands and system limitations.

3.1 High Level Interaction

As most users of the computer music system tend to be computer naive, the interactive graphics approach was adopted which merges on-line terminal aids and visual display correlates. Users are prompted and, wherever possible, not permitted to enter incorrect commands and data. (see below: keyboard commands 4.1).

3.2 Studio Like Operations

The commands and operations of GCOMP are outwardly based on recording studio-like procedures. These may be summarized as follows:

<u>gcomp</u>	<u>studio</u>
amplitude control	faders
location	pan pot
apparent distance	reverb mix
reverb	decay time
filtering	parametric equalization

Digital recording and digitization of pre-recorded material will represent a major part of the work at QUCMF hence we chose these operations to partially simulate typical studio tasks.

The principal difference from the recording environment besides non real-time is that every operation can be made dependant on any other. The idea here is that the 'strips' in an analogue console cannot know and respond to events in other strips except in the rather rudimentary case of automated mixdown. One of the prevailing strategies was then, to provide a degree of 'intelligence' or at least interdependence amongst virtual components of a graphically represented mixing desk.

3.3 Independence From Signal Processing Software

The control structures built graphically in GCOMP are assembled into a 'score'-like time ordered lists. These lists then drive a software synthesis language such as MUSIC-11. This was then tested by running several mixes in stereo. This module is now being modified to generate cmusic compatible lists and could be similarly modified for any software or real-time system.

3.4 Independence From Processing Algorithms

The processing algorithms reside within the music language and therefore, can be modified and 'tuned' by the user independently from GCOMP. For example, the distance function and reverb function built in GCOMP are not bound to any particular acoustical space simulation algorithm realized in cmusic or on a real time device.

3.5 Generality and Portability

GCOMP is written entirely in 'C' and therefore is transportable to most UNIX systems. The output modules that drive the Norpak IGP vector display system have been kept isolated from the internal handling of functions and control data. As stated above, the score module can be readily modified to drive many music systems.

The first installation of GCOMP away from QUCMF will be at the Defense and Civil Institute for Environmental Medicine in Toronto. This site, however, has nearly identical hardware to QUCMF.

4. GCOMP Design

Diagram 2 illustrates the basic modules of GCOMP and their relationship to the user and the sound processing system.

4.1 User Interface

The user interface is divided into four components: keyboard commands, alphanumeric display, graphics display, and sound playback. At this time the playback command is not implemented due to system changes at QUCMF.

Keyboard commands include; on-line help for most operations, two-character command input with the remainder of the command word automatically echoed, logical division of commands into lowercase, uppercase, and control characters, and single RUB keystroke to cancel the current line.

Alpha-numeric display is in fixed position format for full-screen oriented operation. All input is instantly prompted by half-tone character information reducing the dependence on manuals and documentation. Illegal input is not echoed where possible to further minimize keyboard operations. Numeric information is displayed and updated with each command and the user has the options of a verbose mode which displays a longer command history and to view the numeric data as ordered lists. (see diagram 3)

GOOMP: GRAPHICS CONTROL OF MIXING AND PROCESSING

Diagram # 2

Graphics are displayed in the IGP Hewlett Packard 1310A Display screen. Each mode (described below) has a unique format. The Process and Mix formats are time-windowed with user variable durations and interval times. Time-varying control functions may be constructed in Function mode or directly within the other two and are displayed in simple graph form with the exception of the filters which are displayed as pseudo-3D functions. (see diagram 5)

4.2 Modes of Operation

To simplify the specification of acoustic parameter control functions, GCOMP has been divided into three Modes: Function Definition Mode, Processing Mode and Mix Mode. As described above, each Mode has a unique display format and command set. The transferring of data and functions between the modes is accomplished through sets of doubly linked lists which are described below under data structures.

Function Mode

Function definition Mode utilizes an interactive function editor to create general purpose function tables of default-length, 256 floating point numbers, stored in <name> .fn" files. The data structure for functions is a "C" structure:

```
struct fn_type
{char fn_name [ID_SIZE]; name
int count;           number of usages
float vals[FN_SIZE]   function table
struct fn_type *fn_link; pointer to next
}
```

Commands for editing functions include a library of trigonometric functions, arithmetic operations, reverse, invert, join, smooth, raise, lower, and free-hand cursor control (from a graphics tablet or the terminal).

Diagram 4 illustrates a damped sine wave created at the graphics screen by the addition of an exponential curve and a sinusoid. Complex waves of summed sinusoids may also be constructed for use as signal functions in the software synthesis routines.

Process Mode

To describe asynchronous acoustic events, Process Mode constructs data lists, each one describing time varying functions on acoustic parameters. The parameters include gain, stereo location, reverberation, apparent distance, and four equalizing filters (each with gain, centre frequency and band-width controls). There is an extra list forming a

Trigger Track which is used to synchronize events with user-labelled points in time.

Process Mode allows specifications to be made to one sound file at a time, each specification being made up of a sequence of time varying functions for each acoustic parameter to be affected. The functions may be created with the function editor or they may be straight line segments. Diagram 5 shows the layout of the Process Mode screen. Note that the damped sinewave (diagram 4) is now displayed as a channel location control function with duration and starting time as specified in this particular session.

The overall data structure is a set of doubly linked lists with nodes describing the behaviour of the associated acoustic parameters. For each parameter list (gain, reverb, location, etc.) there is header structure, an item in the data list structure, and a time pointer structure.

All process mode data is stored as a binary file, <name>.prc, which can be reloaded during later sessions, or can be used to drive score generating routines producing ascii files, <names>.sc, in a form acceptable to the sound synthesis language. (This assumes that a sound file reading instrument of the appropriate control and processing format has been constructed to accept time-ordered play lists. The module in GCOMP which generates the <name>.sc files can, of course, be modified to suit various music languages).

The window shown in diagram 5 can be changed in duration and can be moved forwards or backwards through the control lists. A cursor locates the item of the parameter list last accessed and may be used to aid insertion or deletion (rather than absolute time declarations). Events may be synchronized by naming items in any parameter list and inserting new items with durations relative to the named items.

Another form of relative time designation may be achieved through the trigger track. This allows synchronization of timings generated by previous sessions or even from timings marked and labelled in the QUICMF Sound File Editor.

Mix Mode

Many component sound tracks may be mixed together under the control of data produced by this mode. As diagram 6 illustrates, only the amplitude and stereo location control functions (both constructed in Process Mode or directly in Mix Mode), are displayed. These two have been chosen to visually represent the primary mixing operations. Entry times for each sound file are specified in relative or absolute durations and, as in the Process mode, the display window may be varied with respect to duration and initial time.

There are six windows displayed vertically at any one time in order to reduce visual congestion. However, there are 40 channels available in total.

6. Evaluation

Separation of the mixing/processing task into these three modes allows the user to build complex control functions for multiple component sound channels without being overwhelmed by the quantity of data being produced. Each mode presents to the user a detailed view of one aspect of the mixing/ processing task, but allows access to only as much data as required. Similarly the graphics display for each mode shows only as much information as necessary to perform the current task.

The interactive nature of the graphics station has been extensively incorporated into the design and implementation of GCOMP. A severe drawback to the system is the time lag between description of the processing and mix and eventual output of samples. It is hoped that the incorporation of a real-time processor will alleviate this problem.

Although both a light-pen and a digitizing tablet have been interfaced to GCOMP we have not yet determined the most efficient and effective mechanisms for user interaction and intend to re-evaluate this as we get more user feedback.

A possible modification to the system is to permit the user to edit the output score as alphanumeric text files which could in turn be turned back into GCOMP and update all previous timing and control data. The nature of the intermediate storage in the form of the .trg, .prc, .fn, and .mx files however made this option a rather expensive luxury. Moreover, this is a graphics process which suggests that operations and revisions are conducted from the interactive graphics mode.

7. Conclusions

Ignoring the rather critical issue of real-time for a moment, the following extensions to a conventional mixing desk have been realised in GCOMP:

- i. all operations are dynamically time dependent
- ii. use of a 'trigger' to automate inter-function coordination
- iii. direct visual correlates to acoustic events
- iv. "virtual" number of input channels
- v. automatic signal correction and amplitude balancing schemes can be imbedded in the processing and mixing algorithms of the software or hardware synthesis system.

Until a real-time implementation can be constructed, it is slightly presumptuous to assume that we have actually improved on the analogue setting. However, more intelligent mixing consoles are beginning to appear and with the development of all digital recording environments, techniques such as those modelled in GCOMP will become the norm.

8. References

- Buxton, W. 1978. "Design Issues in the Foundation of a Computer-Based Tool for Music Composition". University of Toronto Computer Systems Research Group Technical Report: GSRG-97.
- Buxton, W. and Fedorkow, G. 1978. "The Structured Sound Synthesis Project (SSSP) - An Introduction". University of Toronto Computer Systems Research Group Technical Report: SCRG-82.
- Chowning, J. M. 1971. "The Simulation of Moving Sound Sources". Journal of the Audio Engineering Society v2, n6.
- Chowning, J.M., Grey, J.M., Moorer, J.A., Rush L. 1975. "Computer Simulation of Music Instrument Tones in Reverberant Spaces". CCRMA Technical Report STAN-M-1, Department of Music Stanford University.
- Eargle, J. 1976. "Sound Recording". New York: Van Nostrand Reinhold Co.
- Hayes, P. and Reddy, J. 1979. "An Anatomy of Graceful Interaction in Spoken and Written Man-Machine Communication". Department of Computer Science Report Carnegie-Mellon University CMU-CS-79-144.
- Kernighan, B. and Ritchie D. 1978. "The C Programming Language". Englewood Cliffs, New Jersey: Prentice Hall.
- McNally, D. 1979. "Microprocessor Mixing and Processing of Digital Audio Signals". Journal of the Audio Engineering Society v27 n10.
- Moore, F.R. 1980. "CARL" (Documentation of CMUSIC and related programs) UCSD Department of Music.
- Rush, L. (unp). "EDSND Command Summary". CCRMA, Stanford University
- Theile, G. and Plenge, G. 1977. "Localization of Lateral Phantom Sources". Journal of the Audio Engineering Society. v25 n4.
- Tilson, M.D., 1978. "GPAC/IGP Users Manual". Toronto, Ontario: Human Computing Resources, Technical Report.
- Vercoe, B. 1980. "Music 11 Reference Manual". Cambridge Massachusetts: M.I.T. Experimental Music Studio.

GOOMP HELP COMMANDS
=====

(all commands may be accessed from within the current mode)

MIX MODE Commands :

(Upper case indicates the abbreviated form of the command.)

DElete < list filename > : delete process data list from current mix display
INsert < list filename > : insert process data list into current mix display
LIst : list process data lists in current mix display
MOve < listname > : move a process data list to new time and/or window
OVerwrite < mix filename > : write mix data to existing file
REad < mix filename > : display named mix file
TIme < integer > : set start of window display to given time
WIndow < integer > : set window duration to given time

PROCESS MODE Commands :

(Upper case indicates the abbreviated form of the command.)

CLear : clear igp screen
ERase : erase current segment
INsert <linetype> <values> <duration> : insert segment
LIst <parameter> : print list of parameter data
MArk <name> <time> : mark a time on trigger track
NAme <name> : give name to current segment
OVerwrite <filename> : write control data to existing file
PARameter <parameter id> : change current acoustic parameter
REAd LIst <filename> : read control data from disk file
REAd TRigger <filename> : read trigger track from disk file
SEt up <control data> : alter control data parameters
SCore <filename> : produce score file
TIme <integer> : set start time of graphics display
WIndow <op> <integer> : set window size of graphics display
WRite LIst : write data list to disk file
WRite TRigger : write trigger track to disk file

Control Characters :

ctrl C : return to unix
ctrl S : reverse scroll mode

Cursor Controls :

cursor left : move current segment one place to left
cursor right : move current segment one place to right

Change Mode Comands :

M : go to mix mode
F : go to function mode
q : exit program

FUNCTION MODE Commands are:

(Upper case indicates the abbreviated form of the command.)

COsine < frequency > < amplitude > < phase > : display cosine function
Display <function name> : display named function
EXponential < min value > < max value > : display exponential function
FLip : reverse displayed function (about vertical axis)
GRid < parameter > : display grid with appropriate scale
INvert : invert displayed function (about horizontal axis)
LIst < function name> : list stored functions
OVerwrite < function name > : store displayed function under existing name
SIne < frequency > < amplitude > < phase > : display sine function
SMooth : apply smoothing algorithm to displayed function
STore < function name > : store displayed function in main memory
TAg : tag x coordinate of current cursor position
WRite < filename > : store displayed function in disk file

Control Characters

cntrl g : display/delete grid
cntrl o : go to unix
cntrl s : reverse scroll mode
cntrl t : activate/deactivate tags

Graphics Cursor Controls

1 : move cursor left and minimise y coordinate
2 : move cursor down
3 : move cursor right and minimise coordinate
4 : move cursor left
6 : move cursor right
7 : move cursor left and maximise y coordinate
8 : move cursor up
9 : move cursor right and maximise coordinate
SPACE : reverse size of dx

Mode controls

P : go to process mode
M : go to mix mode

Diagram 5 Process Mode

Program 6 Mix Mode