PRECOMP/GT -- A Graphic Tool for Learning Topics in Computer-Assisted Composition

Richard Kaufman
Computer Music Project
School of Music
University of Illinois at Urbana-Champaign
Urbana, IL 61801

ABSTRACT
Visualization of the effects of constraints placed upon variables relevant to the production of a computer-assisted composition (e.g., intervals of intensity & pitch, linear density within time segments) can be useful to both the composer when defining parameters to be used subsequently in the generation of a score as well as the student of computer-assisted composition (CAC) and formalized music. This paper discusses an instructional tool which is designed to meet these needs as well as to demonstrate related topics in computer-assisted composition.

1. Introduction
Computer-based tools for musicians ranging from sound synthesis/analysis programs to compositional programs capable of realizing a complete piece to notational programs devoted to producing high-quality manuscripts are currently available and are under continual development. Many of these programs are intended for use by the computer-literate musician. Preparation of input files, interpretation of results, creation of custom modules to be used in conjunction with the programs are tasks best handled by those who might be called "computer musicians". These programs are primarily concerned with music production, as opposed to music instruction.

For the student of music not familiar with computer science, a need exists for instructional software tools which require a minimal amount of knowledge about the computer itself yet at the same time enable the student to exploit the capabilities of the machine for completing the task at hand. Computationally-repetitive tasks can be automated for the student as much as possible, allowing him to concentrate on high-level aspects and requiring only intermittent direction from the user to influence the final results.

Precomp/GT is such an instructional tool which has been developed at the Computer Music Project at the University of Illinois. It is a computer-based educational tool which employs extensive menu-driven graphic displays and which seeks to provide a general means for instruction about and experimentation with various topics related to the fields of computer-assisted composition and formalized music.

Running on an IBM RT PC workstation using the IBM 5151 Advanced Color Graphics Monitor along with the IBM 6180 Color Plotter, Precomp/GT provides an efficient, friendly means for exploration and experimentation with mathematical models of music to both the student and composer.
Precomp/GT is written entirely in the C programming language using IBM's Graphics Development Toolkit C language bindings [BB 86] and has the capability of interfacing with other, special-purpose packages designed within the Computer Music Project to produce high-quality notation and/or sonic realization. It is intended to be an open-ended system, easily maintained, modified and extended.

This paper briefly describes the design and implementation of Precomp/GT.

2. Characteristics

The "traditional" approach to the design of an educational system implies the ultimate goal of imparting "knowledge" in the form of facts, relationships, the ability to execute algorithms, etc. Tutorials, drill and practice sessions, interactive goal-directed dialogs using artificial intelligence techniques are a few methods which have been used for computer-assisted instruction programs.

Precomp/GT focuses on three aspects of the educational process: demonstration, experimentation and application. The user is presented with a number of modules, each devoted to a specific compositional task. If desired, the student may choose to run a limited demonstration of each topic to familiarize himself with the principles involved. Alternatively, he may choose to conduct experiments with the program, supplying any relevant parameters, altering these parameters at will and continually examining the effects of each change. Finally the student may choose to apply one or more of the modules separately or in concert to obtain a complete, coherent segment of music (within certain limits of the program).

Apart from limited explanations of modules which are available as "help" menus, Precomp/GT does not attempt to "teach" any theory of composition (in the intuitive sense of the word), rather it provides a basis for exploration of concepts and is therefore most appropriately used in conjunction with the direction of an instructor or textbook.

The choice of topics itself represents a highly subjective decision. In its present stage, Precomp/GT centers around principles of music formalization [Xe 71] and pays little or no attention to other directions in contemporary music. However, the program can be expanded to include other topics within the same framework. Suggested extensions to the Precomp/GT include providing higher-quality notation techniques within the program, availability of other probability distributions and improved approximation of values over time through curve-fitting techniques.

Finally, Precomp/GT does not attempt to provide a general-purpose program for producing a work of computer-assisted composition. Modern composers require a much finer degree of control over the evolution of all but the simplest pieces of work and are therefore better served by using more specific programs which are more closely related to their individual needs.

With these points in mind, we describe the structure of Precomp/GT and give an overview of the implementation.
3. Topics in Precomp/GT

When planning the design of a system such as Precomp/GT, the selection of what features to include becomes a matter of "weeding out" those topics which are either:

(1) **Unlikely to be useful to the student of composition.**

Falling into this category are little-used probability distributions, formulas which involve an inordinate number of variables or mathematical models which appear overly-complex to the average student. There is a tendency to attempt to include "everything under the sun" in a project such as this, but a more reasonable approach is to strive for simplicity while allowing for expansion.

(2) **Inordinately computationally-intensive.**

Initially, Precomp/GT was implemented on a machine which did not have a floating-point coprocessor available. As in all interactive instructional programs, response time is important, therefore all tasks which are to be included must be classified as simple to moderate in computational complexity.
(3) Difficult to represent graphically in a clear, comprehensible fashion.

The emphasis is Precomp/XT is on clarity of presentation and ease of interactive modification of musical parameters. Screen space must be reserved for user activity with a mouse and/or string input as well as prompts and "help" text. Viewport boundaries are reserved for menu displays. Simplicity, again, is a desirable goal. A noisy, cluttered display can quickly make "features" into "bugs".

Ultimately, the best judge of which features to include in a program such as this is probably the experienced teacher of topics in composition general and in computer-assisted composition in particular. In effect, we include those topics most likely to be (admittedly subjective) answers to the question: "If you had such a system available in your classroom, what features would be most consistently useful to you and your students?".

It should also be clear that Precomp/XT deals with creative aspects of late 20th-century composition and not with the teaching of common language or more traditional music theory.

3.1. Probability Distributions

Xenakis describes possible uses of stochastic distributions to shape a sequence of musical events [Xe 71]. The composer specifies values for parameters which govern the behavior of event selection. These events, mapped to integer representations, are generated and placed in time either randomly or by the composer according to specified distributions. Precomp/XT provides the user with the means to explore this approach to composition. Currently, continuous probability, Poisson, Gaussian and binomial distributions are available for examination and use. The student may apply the expression to any number of sound parameters (i.e., duration, pitch or frequency, dynamics, timbre) and to allow the system to carry out the computations, producing values corresponding to musical events which obey the specified constraints. Further control may be exercised by applying sieves (see 3.3) to explicitly filter unwanted values. Mean values of parameters may be entered directly by the user, read from a properly-prepared input file or even loosely-approximated by specification of a few control points via a mouse (necessary intermediate values are automatically generated by the use of approximating spline curves [Ba 86]).

The intention of this (the most complex) module of Precomp/XT is to provide a reasonably in-depth method of experimentation for the student in which complete segments of music can be generated in a reasonable amount of time.

Beyond very limited display of randomly-selected values (e.g., display on a "time-line" of durations), Precomp/XT does not attempt to produce high-quality on-screen notation of the results, although the capability is easily obtained (see below).

3.2. Vector Space

As a means of visualizing, defining and manipulating qualities of sonic events, 3-dimensional vector space methods [Xe 71] are useful to the student of composition. These methods, which define an event in terms of intervals and qualities (e.g., pitch, duration and intensity), along with algebraic operations which can be performed upon the vectors are available within Precomp/XT. The program provides a 3-dimensional "window" in which the user can actually "see" and
experiment with the vectors, defining and manipulating them along with the context in which they are viewed. Standard 3-dimensional transformations are available to the user, including scaling of axes and objects and full 3-D rotations of the display. Also a stand-alone portion of the program, this facility provides an attractive alternative presentation of the topic.

### 3.3. Sieve Theory

This method of specifying constraints or restrictions upon acceptable values for musical parameters is available for experimentation within Precomp/GT. Primarily used for expressing simple and complex scales, sieve construction ([Hh 59] & [Ke 71]) involves determining modulo classes and combining them with the standard boolean operations AND, OR and NOT. Precomp/GT provides the user with the ability to specify, modify and display sieve elements. Three additional operators are provided: XOR (exclusive-OR), NAND and NOR. These operators are defined in terms of the basic three and may serve to simplify lengthy sieve expressions thereby encouraging further exploration. For example, Xenakis' sieve expression representing the major scale:

\[
( \bar{e}_2 \cap \bar{e}_6 ) \cup ( \bar{e}_3 \cap \bar{e}_4 ) \cup ( \bar{e}_2 \cap \bar{e}_4 ) \cup ( \bar{e}_3 \cap \bar{e}_4 )
\]

may be represented more simply as:

\[
( \bar{e}_3 \text{XOR} \bar{e}_4 ) \cup ( \bar{e}_4 \text{XOR} \bar{e}_5 )
\]

The implementation of the sieve module of the program is made simple, elegant and extendible by the use of the Lex lexical analyzer generator [Le 75] and Yacc parser generator [Yo 75] tools which are available as part of the Unix operating system. A formal grammar is specified for sieve expressions and the user's specifications are tokenized and parsed internally prior to graphic display. Although primarily intended for sieving of pitches, Precomp/GT does not limit the sieve construction process to this parameter. In fact, use of sieves can be combined with experiments in probability distributions (see 3.1) to produce limited segments of music.

### 3.4. Tone-Rows

When dealing with tone-rows, the student follows an algorithmic approach to manipulating matrices composed of pitches within the scale. This can be a tedious and error-prone process for the student. In a separate module, Precomp/CT automates a few basic processes for the user. Essentially a stand-alone portion of the program, specification can be done either by text input in musical notation (entering "A\#", "F", "G-", etc.) or by using a mouse to indicate the physical position of each note on a musical staff displayed on the screen. In either case, the student can request display of the original tone-row, retrograde and inversions while continually adding, deleting and moving notes in the tone-row. Finally, hard-copy results of the tone-row matrices can be produced from within this module.

In its present stage, this part of Precomp/CT includes only a few rather simple serial techniques. However, this is a foundation upon which more sophisticated procedures (such as combinatoriality) can be built.
As mentioned at the beginning of this section the facilities currently available within Precomp/GT do not represent the full spectrum of "useful" utilities which could be included in such a project, but a starting point developed during the spring and summer of 1987. Work continues on defining new modules and refining existing ones.

4. Graphic Capabilities/Implementation

The preferred way to obtain results of the realization of a composition is through a standard manuscript and/or sonic realization of the piece. Although Precomp/GT is not intended to function as a notation program, the results obtained can be processed by other programs available at the Computer Music Project. Used in this manner, Precomp/GT may be logically viewed as part of a pipeline of specialized music programs.

For standard notation, the user of the program can indicate that the results of experiments are to be saved in a file which is suitable for submission to Orpheus [Cô 87], a music notation program currently available on an IBM PC/AT system at the Project. Slight inconveniences due to the differing operating systems employed by the two machines may be overcome by file translations. The manual intervention required is not overly difficult. Work continues on porting Orpheus to the ET system, which will eliminate this difficulty.

Finally, sonic realization of the results of experiments with Precomp/GT will be possible through the use of the m4c [Au 85] sound synthesis program. By piping results from Precomp/GT to this program, musical segments can be played back within a reasonable time. Limitations include the specification of "instruments" which m4c requires and the mapping of Precomp/GT internal representation of note vectors to a form suitable for m4c. Further advantages can be gained by eventually making Precomp/GT output available in a format suitable for the Notepro preprocessor. We hope to implement these features of Precomp/GT in the near future.

5. Conclusions

Work to date on Precomp/GT represents a starting point for a general computer-based aid in the teaching of computer-assisted composition. From the viewpoint of the user, simplicity of design and the graphic user-interface provide a pleasant way for exploring the capabilities provided. From the developer's perspective, the use of a highly modular design, the use of high-level software development tools and the attention to implementation of graphic displays in an accepted "standard" (GKS-style function calls) will provide long-term benefits in terms of maintenance, modification and extension. As in all such development projects, feedback from the target audience, namely students of composition, is anticipated as the final judge of the success of the approach.

Acknowledgments

Many thanks are due to the staff of the Computer Music Project for continued advice and assistance during the development of Precomp/GT, and to Prof. Sever Tipei of the U.I School of Music whose assistance throughout each phase of the project development is greatly appreciated.
References


