1. Abstract

This paper describes AREM / ARMS, a computer program that applies in real time some conceptual tools of musical analysis. Algorithms for pitch and rhythm reinterpretations are supported by the program; it also acts in terms of texture and syntax. The program can be applied in interactive performing, computer-aided composition and musical teaching.

2. Introduction

AREM / ARMS (Algoritmos para la Reinterpretación de Estructuras Musicales i Algorithms for the Reinterpretation of Musical Structures) is an interactive program which allows the transformation of musical material input by a performer or composer. Musical categories (pitch, rhythm, texture and syntax) for input reinterpretation involve analytical tools related to a composition method in usage (Francisco Kröpf’s theory of musical analysis and composition), and further compositional decisions (inside or outside the program) are subsidiary of this method. The program is applied to live-electronics performing, computer-aided composition and musical teaching.

The modularity of the program and graphic edition capabilities allow a non programmer performer or composer to construct complex musical algorithms based on simpler ones. The input is performed on a MIDI controller/synthesizer attached to a Macintosh computer running the MAX programming environment.

The first part of the program uses algorithms to transform the pitch and rhythm inputs [Calzón, 1992]. Here we introduce the aspects concerning texture and syntax reinterpretations.

3. Reinterpretation in terms of texture

We take into account, for the purposes of this program, three types of single textures: monody (linear, horizontal), polyphony (superimposed monodies), and homophony (relating to chords, vertical).

There are several algorithms in AREM / ARMS that produce transformations on the input as follows:

3.1 monody -> polyphony; two or three rhythmic simultaneous transformations are superimposed.

The several polyphonic layers are variations of the original monody applying these methods:

* triggering at different times
* transposition at different intervals
* application of different rhythmic reinterpretations (AS, AR, TS, TR - agogic, dynamic) [Calzón, 1992]
3.2 **monody -> homophony**: horizontal segmentation of the monody in terms of three, four or five elements. These groups are verbalized by one of these methods:
* loop the original pitch registers in the monody
* octave transposition of the pitch classes, in order to change the full vertical extension of the chorû
* transposition in an interval, preserving the intervallic chord structure.

3.3 **homophony -> monody**: an arpeggio of the input chord is produced. The order of the pitches depends on the microtemporal delay between simultaneously played notes because of MIDI sequential transmission. This short delay is expanded by the program.

3.4 **homophony -> polyphony**: de-phasing of the simultaneous voices of several input chords producing a polyphonic effect. One or more split points are set in the keyboard, vertically segmenting the chord. Each segment in the chord is processed separately as parallel monodic inputs, as seen in 3.3

Depending on the type and magnitude of the applied transformation, it is possible to control the similarity between input and output. For example, working with identical pitches will produce a transformation nearest to the input. We call this quality the degree of similarity of the reinterpretation.

4. Syntactical reinterpretation

The activation of a pedal or a certain time lapse marks segments in the performed input [Calzón, 1992]. Each segment is symbolized by a letter (A, B, C ...). When the order of the output symbols differs from that of the input symbols, the reinterpretation is considered syntactic. The internal configuration of each symbolized segment is shown between square brackets. For instance, AB=[ab][ce] means that the first segment (A) has two different components (a and b), and the second segment (B) repeats the same component (c) twice. The square brackets also indicate a perceptible segmentation between the second and the third lower case symbol.

As the possible number of combinations of symbols is extremely large, a system that "prunes the combinatory tree" has been incorporated in order to produce only meaningful sequences in contemporary music, avoiding trivial and disordered combinations, such as:

\[ AA=[[aa][aa]] \]
\[ AB=[ab][cd] \]

The syntactic qualities of the symbolic chains that are not discarded are classified according to the general categories of permanence/change. Four basic modes of symbol sequences are considered by the program: reinventive, in period, evolutive and contrastive. When the first symbol in capital letters is A, the following symbol can be A, A', A* or B.

4.1 **Reinventive syntax (AA)**: a reinvention is always a closed phrase.

selected:  \[ AA=[ab][ab] \]
discarded: AA=[aa][aa]

4.2 In period syntax (AA*): A* represents a segment that is a complement of A (a variation). A* contains at least one symbol included in A and ends as it began (low level symmetry, closed phrase).
selected: AA*=[ab][ba] or [ab][ca]
discarded: AA*=[aa][ba] or [ab][aa]

4.3 Evolutive syntax (AA'): A' represents a segment that grows from A (a variation). A' contains at least one symbol included in A and ends differently from the way it began (low level asymmetry, open phrase).
selected: AA'=[ab][ac] or [ab][bc] or [ab][cb]
discarded: AA'=[ab][bb] or [aa][ab]

4.4 Contrastive syntax (AB): a contrast is always an open phrase.
selected: AB=[aa][bb] or [aa][bc] or [ab][cc]
discarded: AB=[ab][cd]

As an heuristic rule in a four symbol sequence we consider that a symbol must not appear more than twice, and the sequence must have at least one repeated symbol. In order to create an internal segmentation, a short delay is added between the second and third segment of the chain. Similar considerations are made for five elements chains (2+3 or 3+2).

The user sequences three segments, each of them associated with a symbol. Then he or she chooses one of the above syntactic modes, and the program combines the sequences according to the selected syntactical quality.

5. Modularity
Each reinterpretation algorithm is represented as a MAX box-object controlled by pop-up menus. Other box-objects represent input/output conditions, for instance:
* segmentation using a pedal or time lapse
* selection using mouse, computer keyboard, triggering notes or remote switches
* MIDI input or graphically edited pitches

The user may construct patches where the transformations act on one or more of the parameters considered above (pitch, rhythm, texture, syntax) according to a composition or improvisation plan.

6. Acknowledgments
I would like to thank to Francisco Kröpfl for providing the theoretical basis of this paper, and Lorraine Smith for helping to prepare its English version.

7. Reference