Parsing real-time musical inputs and spontaneously generating musical forms: Hierarchical Form Generator (HFG)

David Rosenboom

Center for Experiments in Art, Information and Technology, (CEAIT), School of Music, California Institute of the Arts, Valencia, CA USA 91355; email: davidr@calarts.edu

ABSTRACT

Software for parsing real-time musical inputs into musically meaningful sub-groupings, such as phrases or gestures, for the purpose of extending the formal possibilities of interactive performance is described in brief outline. The parsing algorithm is based on a partial model of musical perception emphasizing analysis of musical contours in several parameters.

1. BACKGROUND AND MOTIVATION

1.1 Relation to earlier work on brain activity

This is a practical application ensuing from earlier research in which it was necessary for a computer to be able to identify events in an unfolding musical fabric that might be perceived by a listener as having important structural significance. Once identified, concomitant electrical brain activity could be studied. The intention here is to utilize this musical boundary detector as a tool in building an intelligent responding system for live performance involving improvisation. For more extensive technical detail, consult (Rosenboom 1990).

1.2 Special considerations for structured improvisation - alternative to score following

Except for a brief, recent period in Western high-art music, the importance of structured improvisation in the evolution of musical styles has been recognized across history and world cultures. Its significance is now being reiterated in Western culture as a fundamental musicianship skill. I believe the cognitive organizing schema involved in listening to music with unknown and unpredictable content to be different than in music with known content and/or familiar language elements. Most music mixes the familiar and unfamiliar in an essential tension. Consequently, the requirements for an intelligent, responding instrument for improvisation are different than for traditional score following in which the computer already *knows* the score. The system must also operate on various levels of musical form in addition to creating immediate responses to changes in acoustic parameters. This approach presents an alternative to traditional score following and may contribute useful techniques for alternative score styles in which performer choices are significant.

2. PARSING SYSTEM

2.1 Assumptions for intelligent reaction to unpredictable musical inputs

1) A parsing mechanism is needed to automatically isolate meaningful musical units, termed *subshapes*, so they can be manipulated and transformed. 2) The musical input to be parsed is essentially unpredictable.

2.2 Form of input

Inputs consist of parametric *shapes* on low and high levels of perceptual organization, see Fig. 1. Some parameters, e.g. pitch and loudness, require only tracking changes in time. Others, e.g. timbre, harmonic distance and rhythmic pattern similarity, require multi-dimensional space mapping with appropriate metric, (e.g. Euclidean, Minkowskian, "city-block", etc.). Not all relevant parameters can be obtained through the MIDI data stream. However, much can be accomplished with just MIDI pitch and velocity, see Fig. 2.

2.3 Parsing engine principles - partial model of perception

Decisions are made about musical boundaries using difference detection and estimation of *initiator strengths*. The latter is determined by comparison with a threshold being updated in time according to contextual variance and short-term memory decay. Results from several parameters are weighted and tested for *minimum strength* required to initiate a boundary. Parametric weights are adjusted empirically to closely match perception. In performance, control weighting is a powerful means of directing the responding system towards different results, adding to its effectiveness as a new kind of instrument. Various difference detectors may be employed. Refer to (Rosenboom 1990) for description of that used here. This parsing engine reflects certain functions of core sensory processing elements. No consideration of ***musical content*** is involved. Nevertheless, experience in performances has shown the system to be remarkably adaptable to stylistic features.

3. RESPONDING MECHANISM

3.1 Playback of subshapes

A simple responding strategy is often the most effective, enable the performer to trigger playback of subshapes at will from the input device. A skilled improviser can quickly learn to select new material to play into the parser and trigger playback of old material with fluidity. A key factor in designing an HFG performance is how playback of the subshapes is mapped onto the MIDI input device. The default is to distribute them over the instrument's MIDI note range.

3.2 Transformations and categorical memory organization

Subshapes may be transformed prior to replay. In one example, successive rhythmic patterns are compared to produce an *index of similarity*. This is used to choose from a repertoire of transformations that are rank ordered according to the *degree of variation* they produce (Rosenboom with Sankaran 1990), see Fig. 3. Many other strategies can easily be devised. Similarly, various methods for shape comparison may be applied to group subshapes according to common or related features. These may be replayed and/or transformed together.

3.3 Hierarchies, feedback and self-organization

HFG works with hierarchically organized musical forms in which contours result from tracking global properties, or *states*, of sub-units in higher-level sequence groupings, see Fig. 4. These sequences are generated by simulating another aspect of perception and cognition, anticipation of patterns. The algorithm makes predictions regarding the significance a listener might attach to the occurrence of particular units at specific points in a sequence. Significance, in this case, refers to the *strength* with which a particular occurrence may influence the formation of pattern groupings in perception. Feedback, from musical performance elements or operator cues either confirms or denies the algorithm's predictions. This influences the organization of subsequent pattern groupings. See (Rosenboom 1990) for a detailed example.

4. HMSL IMPLEMENTATION

4.1 HFG performance controls

HFG was created with HMSL (Hierarchical Music Specification Language). The procedures discused above are well supported in HMSL. An example HFG control screen is shown in Fig. 5. Parametric shapes can be input through MIDI recording, (MIDI RECORD, STOP, PLAY), drawing and editing with the HMSL Shape Editor (not shown), or by composition algorithms. A selector control, Parse Shapes, identifies a shape to be PARSEd into SUBSHAPES. Difference detector behavior can be viewed in a Shape Editor window. Min-S slider sets the minimum strength initiators must achieve to cause subshapes capturing. L/P-W determines relative weights given to loudness (MIDI velocity) and pitch. PARSE INPUT turns real-time MIDI parsing on and off. PT and LT are indicators showing pitch and loudness threshold changes in time. Exponential decay time constant is set by TC. T-Decay sets delay interval between updates in the decay process, used to speed up or slow down the system in performance. All on-screen controls can be driven by MIDI. This facilitates tuning the system for different playing styles. For example, low Min-S usually results in many short subshapes being captured because small differences are sufficient to cause boundary determination. Higher values usually result in fewer, longer subshapes. Decay time constants are important in determining how large changes mask effects of lesser changes that follow. PLAY HFG turns on SUBSHAPES playback triggering.

4.2 Additional features under development

These include: increased intelligence in scaling changes according to contextual variance over long periods of time, extracting and comparing rhythmic patterns, more hierarchical levels, various shape comparison measures, other prediction and feedback algorithms, link to real-time DSP synthesis, more transformations ordered as to complexity, handling simultaneous events like clusters and chords, segmentation of parallel streams, recognition of left and right hand events for keyboards -- a feature implemented in the Touché computer keyboard instrument by Rosenboom and Buchla in 1979 --, parsing on note durations, and playback dynamics.

5. PROJECTS USING HFG TO DATE

5.1 Compositions and performances

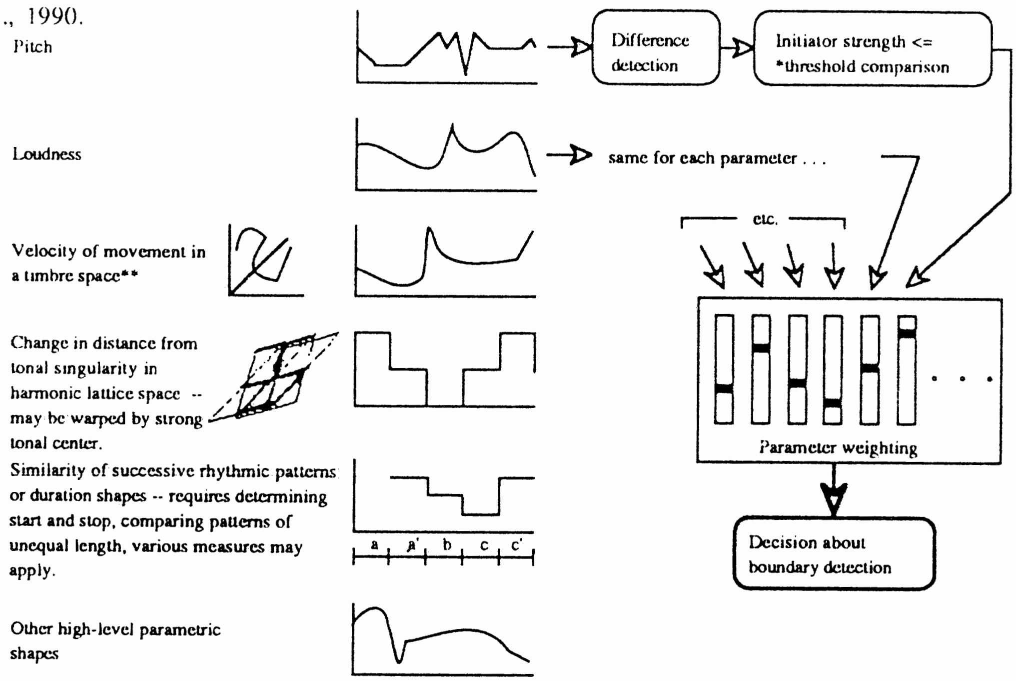
1) A work for Yamaha MIDI grand piano, Disklavier and computer by Rosenboom, *Predictions, Confirmations and Disconfirmations*, (working title), 1991. 2) A work for South Indian Mrdangam and computer music system by Rosenboom with T. Sankaran, *Layagnanam*, 1990. 3) An *Extended Trio*, fully interfaced ensemble including 1) and 2) with jazz bassist, C. Haden, 1992, Fig. 6. 4) A series of solo instrument works in progress by Rosenboom, Fig. 7.

5.2 HFG as a tool for music learning

HFG can produce a spontaneous segmentation analyses of improvised material. This can help performers understand how such material can be heard and organized. HFG is also useful in studying phrasing and interpretation. A musical phrase recognizer has been constructed to help young children identify musical elements contained in spontaneously created material and feed these into a program designed for learning about composition.

6. REFERENCE

D. Rosenboom, *Extended Musical Interface with the Human Nervous System*, Leonardo Monograph Series, International Society for the Arts, Sciences and Technology (ISAST), 672 South Van Ness Ave., San Francisco, CA, U.S.A., 1990.



*Threshold is updated in time according to contextual variance and short-term memory decay.

**Future, real-time DSP may permit continuous parameter updates as in the analog, "old days".

Figure 1 - HFG - Abstract Case

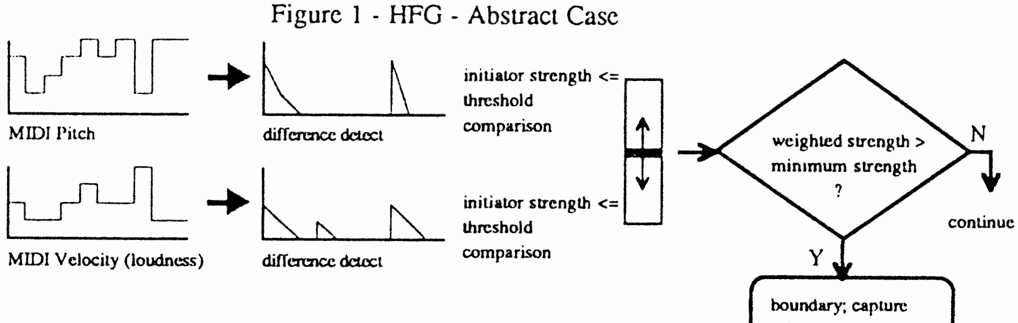


Figure 2 - Practical MIDI Case

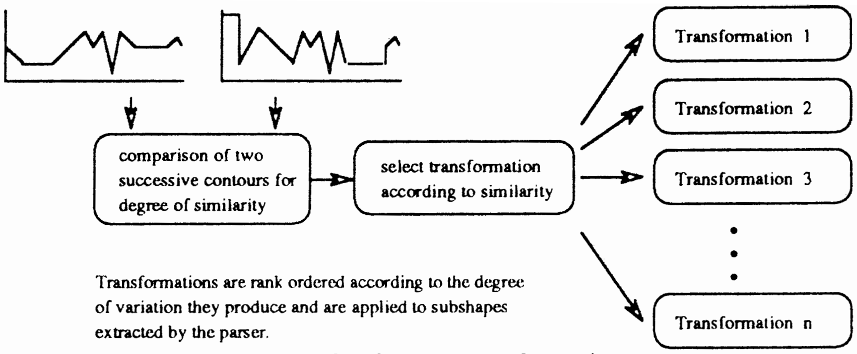


Figure 3 - Output Transformations

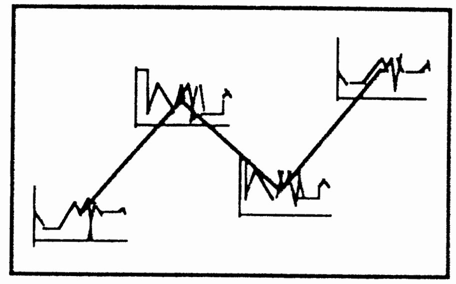


Figure 4 - Contour in Parametric States

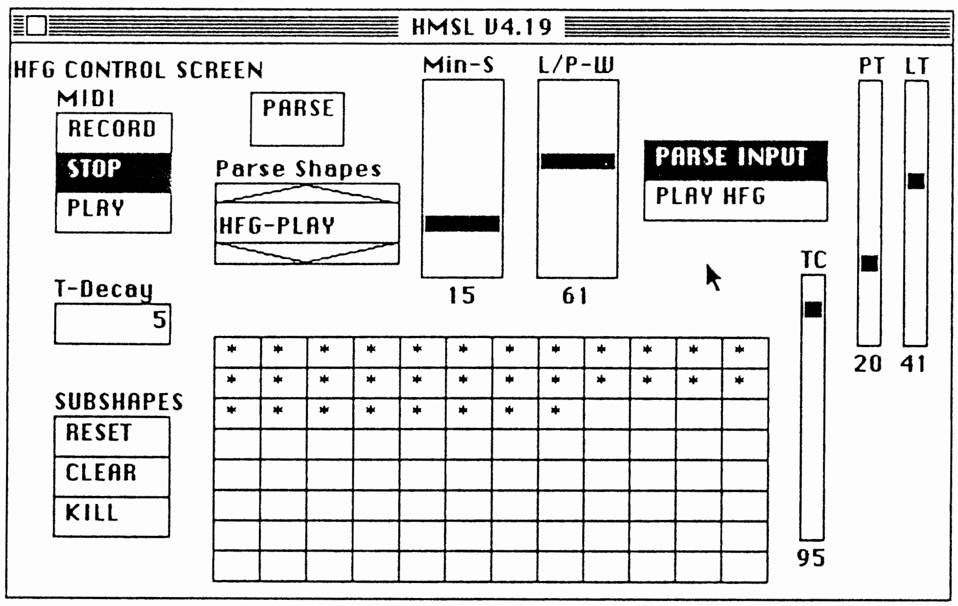


Figure 5 - HFG Control Screen

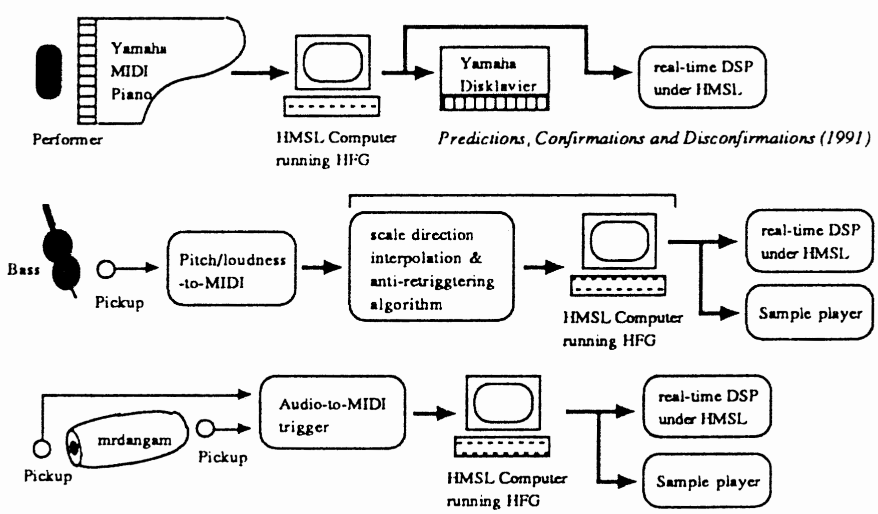


Figure 6 - *Extended Trio (1992)*

Layagnanam (1990)

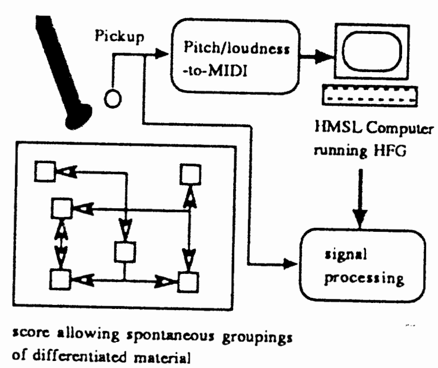


Figure 7 - Scheme for Series of Solo Works