MuSA.RT: Music on the Spiral Array . Real-Time *

Elaine Chew† and Alexandre R.J. François
Integrated Media Systems Center
University of Southern California, Los Angeles, California
{echew,afrancoi}@usc.edu

ABSTRACT

We present MuSA.RT, Opus 1, a multimodal interactive system for music analysis and visualization using the Spiral Array model. Real-time MIDI input from a live performance is processed, analyzed and mapped to the 3D model, revealing tonal structures such as pitches, chords and keys. A user can concurrently navigate through the Spiral Array space using a gamepad or set the camera control to automatic pilot. The interaction among and concurrent processing of the different data streams is made possible through the Modular Flow Scheduling Middleware.

Categories and Subject Descriptors

H.5.5 [**Information Interfaces and Presentation**]: Sound and Music Computing—*Systems*; I.5.5 [**Pattern Recognition**]: Implementation—*Interactive Systems*; J.5 [**Computer Applications**]: Arts and Humanities—*Performing Arts*

General Terms

Algorithms, design, experimentation, human factors, theory

Keywords

Spiral Array, SAI, MFSM, music analysis, system implementation, music visualization

1. INTRODUCTION

A defining feature of tonal music is the unfolding of pitch structures over time. Real-time tracking of tonal patterns in music has widespread applications in music analysis, information retrieval, performance analysis and expression synthesis. Each piece of music consists of a sequential arrangement of notes that generates pitch structures over time. An expert listener is able to ascertain the keys and harmonic patterns traversed over time. But a novice or a computer would benefit greatly from a geometric model that can provide visual cues and numeric quantifying of these tonal properties. We present MuSA.RT (Music on the Spiral Array . Real-Time), a system for real-time analysis and interactive visualization of tonal patterns in music.

Figure 1: The MuSA.RT System.

MuSA.RT maps real-time MIDI input, for example from a live performance, to the Spiral Array [1], a 3D model for tonality. The analysis and graphical rendering reveal the presently active set of pitch classes, and higher level constructs, such as the current chord and key. At the same time, the user can control the camera using a gaming device and navigate through the Spiral Array space.

2. THE SPIRAL ARRAY

The Spiral Array model is a geometric model for tonality rooted in the theory and perception of music. Tonality exhibits a high degree of symmetry and transformational invariance. For this reason, spiral and toroid structures are particularly well-suited to representing tonal relations.

The Spiral Array model incorporates multiple levels of descriptions by generating an array of spirals each representing entities at a different hierarchical level. The outermost pitch spiral has a configuration that clusters pitches

*This project was funded in part by the Integrated Media Systems Center, an NSF ERC, Cooperative Agreement No. EEC-9529152 and an NSF MRI Grant No. EIA-0116573. Any Opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the National Science Foundation.

†USC Daniel J. Epstein Department of Industrial and Systems Engineering

MM'03, November 2–8, 2003, Berkeley, California, USA.
ACM 1-58113-722-2/03/0011.

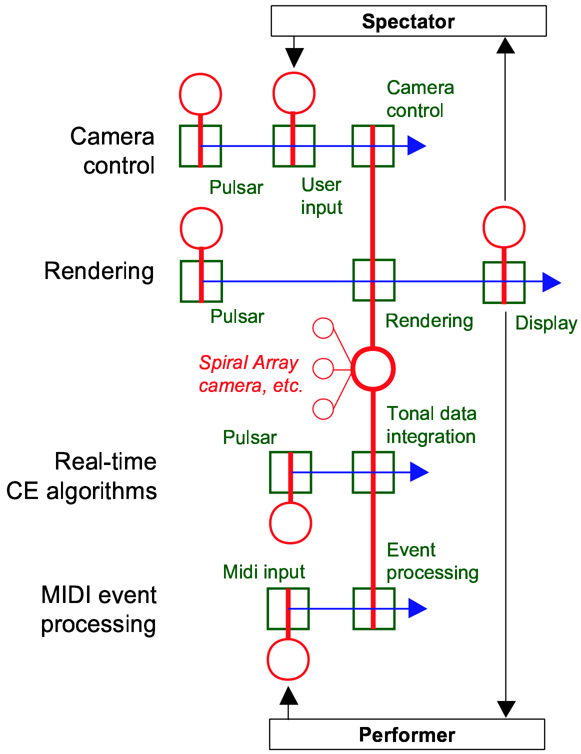

Figure 2: MuSA.RT application flow graph.

that form higher level structures such as chords and keys. Higher level entities are generated as convex combinations of their lower level components. Distinct from other spatial and graph representations of inter-pitch relations, the Spiral Array represents entities from all levels in the same 3D space for comparison and visualization.

The Spiral Array serves both as a visualization and analytical tool for tonal music. The Spiral Array has been shown to be an effective tool for assigning context-appropriate pitch spellings to MIDI numbers [3], for chord tracking [1] and for key-finding [2]. Its nearest neighbor entity from each of the spirals to the CE reveals the appropriate pitch spelling, and its chord and key memberships.

We use the Spiral Array model to quantify, analyze and visualize tonal patterns. A MIDI stream consists primarily of pitch onset and offset events. The MuSA.RT system maps MIDI pitch numbers to their appropriate spellings and corresponding positions on the Spiral Array. The onset and offset times for each pitch is synchronized with the appearance and disappearance of the appropriate sphere at the end of the radial pin connected to its pitch representation.

Each pitch event also contributes to a *center of effect* (CE), a fundamental concept underlying the Spiral Array model. The CE maps any pitch collection to a spatial point and any time series of notes to meaningful trajectories inside the Spiral Array. If $\{p_{ti}\}$ is the set of active pitches at time t, then the CE is given by $c_t = \alpha \sum_i p_{ti} + (1-\alpha).c_{t-1}$.

Contextual information is summarized by the CE; and, the appropriate pitch spelling and the key and chord structures are determined through a nearest neighbor search in the Spiral Array space. The CE of the MIDI stream is represented as a green sphere with a tail that shows its trajectory over the recent past. The active triad is indicated by a colored triangle, red for major triads and blue for minor triads. The active key is shown as a throbbing star at the position of the key representation.

3. SYSTEM INTEGRATION

We designed the MuSA.RT system using the SAI architectural style [4], and implemented the MuSA.RT, Opus 1, prototype using the open source Modular Flow Scheduling Middleware (MFSM, mfsm.SourceForge.net). Figure 2 shows the integrated application flow graph.

There are numerous challenges to visualizing a 3D model on a 2D screen. In MuSA.RT, we overcome many of these problems by allowing the user to dynamically and concurrently control the camera using a gaming device. The user can zoom in and out, tilt the viewing angle and circle around the spiral to get a better view of the tonal structures. In addition, an automatic pilot option will seek the best view angle and center the camera at the heart of the action.

A fundamental idea in SAI is the distinction between volatile data flowing in streams (such as, the video frames for visualization) and persistent data held in shared repositories (for example, the Spiral Array). The system consists of four independent data streams: (1) MIDI input and event processing; (2) tonal analysis (real-time CE algorithms); (3) rendering of the Spiral Array structures; and, (4) control device (gamepad) input and camera manipulation. These four streams potentially operate according to different modalities (push or pull) and at different rates. The four streams interact through persistent yet dynamic structures and through the user(s) to form interaction loops.

4. PERSPECTIVES

MuSA.RT incorporates modules for both interactive music visualization as well as real-time music analysis. The proximity of closely related musical entities makes the Spiral Array a useful tool for visualizing tonal structures and testing algorithms for music analysis. Note that the system is not restricted to the CE method for chord and key tracking. MuSA.RT provides a generic platform for testing and validating algorithms for real-time music tracking.

Future research include: using MuSA.RT to visualize tonal trajectories and inspire algorithms for similarity assessment; exploring other artistic ways to render music information in the Spiral Array space; and, incorporating different modalities for interacting with the 3D space.

5. REFERENCES

[1] E. Chew. *Towards a Mathematical Model of Tonality.* Ph.d. thesis, Massachusetts Institute of Technology, Cambridge, MA, 2000.

[2] E. Chew. Modeling tonality: Applications to music cognition. In *Proc. 23rd Annual Meeting of the Cognitive Science Society*, Edinburgh, Scotland, 2001.

[3] E. Chew and Y.-C. Chen. Mapping midi to the spiral array: Disambiguating pitch spellings. In *Proc. 8th INFORMS Computer Society Conference*, pages 259–275, Chandler, AZ, 2003.

[4] A. R. J. François. Software Architecture for Computer Vision. In G. Medioni and S. Kang, editors, *Emerging Topics in Computer Vision.* Prentice Hall, 2004.