SICIB: An Interactive Music Composition System Using Body Movements

Roberto Morales-Manzanares,[*]
Eduardo F. Morales,[†] Roger Dannenberg,[‡]
and Jonathan Berger[§]
[*]LIM Laboratorio de Informática Musical
Universidad de Guanajuato, Guanajuato, Gto. México
roberto@quijote.ugto.mx

[†]ITESM, Campus Morelos
Temixco, Morelos 62589 México
emorales@campus.mor.itesm.mx

[‡]School of Computer Science
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213 USA
roger.dannenberg@cs.cmu.edu

[§]Center for Computer Research in
Music and Acoustics
Stanford University
Stanford, California 94305-8180 USA
brg@ccrma.stanford.edu

Traditionally, music and dance have been complementary arts. However, their integration has not always been entirely satisfactory. In general, a dancer must conform movements to a predefined piece of music, leaving very little room for improvisational creativity. In this article, a system called SICIB—capable of music composition, improvisation, and performance using body movements—is described. SICIB uses data from sensors attached to dancers and "if-then" rules to couple choreographic gestures with music. The article describes the choreographic elements considered by the system (such as position, velocity, acceleration, curvature, and torsion of movements, jumps, etc.), as well as the musical elements that can be affected by them (e.g., intensity, tone, music sequences, etc.) through two different music composition systems: Escamol and Aura. The choreographic information obtained from the sensors, the musical capabilities of the music composition systems, and a simple rule-based coupling mechanism offers good opportunities for interaction between choreographers and composers.

The architecture of SICIB, which allows real-time performance, is also described. SICIB has been used by three different composers and a choreographer with very encouraging results. In particular, the dancer has been involved in music dialogues with live performance musicians. Our experiences with the development of SICIB and our own insights into the relationship that new technologies offer to choreographers and dancers are also discussed.

Computer Music Journal, 25:2, pp. 25–36, Summer 2001

Background

In 1965, John Cage, David Tudor, and Merce Cunningham collaborated on *Variations V*, a multimedia work in which dancers triggered sounds each time they were positioned between one of a dozen photoelectric cells and a light activated each cell. This revolutionary work challenged notions of the traditional relationship between choreographer and composer—a relationship characterized by compromise of one artist in order to fit the work of the other.

The Cage–Cunningham collaboration achieved equality by transferring control from both composer and choreographer directly to the dancer. Recent technologies offer more refined tools for mapping dancer's movements to music. Computer processing of data from motion sensors allows both composer and choreographer to retain control and achieve an integrated work that is not improvisatory. These systems provide any degree of control desired by the composer-choreographer team; they can also provide the dancer with the flexibility of performer nuance and inflection usually associated with musicians rather than dancers.

The motion detection technology can be classified according to the location of the sensors and detectors. These include sensors and detectors attached to the body of the dancer. Examples of such devices include piezo-electric and flex sensors (e.g., Van Raalte, 1998). Another type of technology involves sensors and detectors placed external to the dancer's body (cameras, infrared sensors, etc.) (e.g., Rokeby 1999). A third type of motion detection technology includes sensors attached to the body and detectors placed strategically elsewhere. These devices include electromagnetic sensors, sonar, etc. (e.g., Morales-Manzanares and Morales 1997).

As is often the case with integrated technologies, newfound freedom is often paradoxically accompanied by stifling restrictions. Researchers, musicians, composers, choreographers, and dancers are just beginning to grasp the possibilities these new technologies offer. However, there remains a distinct need for a simpler, easier to understand, and powerful coupling mechanism that mediates between sound and motion.

This paper describes Sistema Interactivo de Composición e Improvisación para Bailarines (SICIB) designed with the above ideas in mind and capable of generating music in real time based on body motion. SICIB receives space coordinates from sensors attached to dancers and obtains choreographic information from them. Combinations of such information are used to satisfy conditions of "if-then" rules whose actions affect musical elements. SICIB can use two different music composition systems: Escamol (Morales-Manzanares 1992) and Aura (Dannenberg and Brandt 1996).

Escamol is a language for creating scores using grammatical rules (Allen 1987). Aura is an object-oriented software foundation for building interactive systems. In SICIB, we use Aura to create software instruments.

Continuous polling of the sensors allows us to define expressive choreographic information, while the flexibility of the musical systems facilitates expressive musical events. The integration between both is simplified by a rule-based system with a simple syntax. With SICIB, a composer can define regions in space, use information of the curvature and torsion of the dancer movements, and apply continuous changes of a sensor's data to different musical parameters.

With SICIB, choreography does not need to be adjusted to a fixed, pre-determined musical piece: dancers can freely improvise their movements and map these improvised gestures to music generated in real time. SICIB can also be considered a virtual instrument in which music is produced through body movements, opening new possibilities for music performance and composition. Finally, SICIB allows musicians and dancers to interact and improvise dialogues during the course of the performance.

Related Work

Many systems have been proposed that produce music from body movements. They basically differ on the type of sensors used and on their musical capabilities. Winkler (1995) presents an overview of gesture and motion sensing for music making. Systems which use sensors external to the body (such as cameras) detect—in general—changes from continuous frames (e.g., Rokeby's Very Nervous System) or changes between the current frame and a frame of reference (e.g., STEIM's BigEye).

BigEye, from the Studio for Electro-Instrumental Music (STEIM) in Amsterdam, is a computer program that takes real-time video information and converts it into MIDI messages. In BigEye (STEIM 2000), the user configures the program to extract objects of interest. The position of the objects is checked against user-defined zones. MIDI messages are generated each time an object appears or disappears in a zone or moves within a zone.

These events can cause notes to be switched on or off, for instance. Other MIDI messages can be generated using some of the object's parameters, such as position, speed, and size. A camera focused on a choreographer can follow (with sound) the paths of up to 16 individual dancers.

Another related system, the Very Nervous System (Rokeby 1999), detects any movement within a defined active performance area. Position and motion information is used to trigger sounds in real time. The image analysis is focused on motion rather than color or shape information. Each video frame is compared to the previous one to determine what is moving.

In general, systems based on cameras are very good at detecting global changes of the performer (e.g., crossing of regions or changes in overall velocity of the dancer), but they are poor in detecting more detailed information (e.g., which part of the body was responsible for changes in the image). Image processing algorithms capable of detecting more detailed information (e.g., Maes et al. 1995) demand a high processing cost and rely on images with high contrast and a restricted range of orientation with respect to the camera. Another potential problem with camera-based systems involves changes in illumination. Sensors attached to the body are immune from this problem. Still, the simplicity and non-obtrusiveness of camera-based sensing is attractive. Interesting progress has been achieved by the EyesWeb project (Camurri et al. 2000).

Not all external sensors are cameras. For example, floor sensors have been created using various technologies (Johnstone 1991; Pinkston, Kerkhoff, and McQuilken 1995; Paradiso et al. 1997; Griffith and Fernstrom 1998).

Other systems use sensors such as piezo-electric material in the body of the dancer. For instance, Control Suit, constructed at the Norwegian Network for Technology, Acoustics, and Music (NoTAM) in 1995 is a suit equipped with sensors, and the performer produces sounds by tapping his or her own body (NoTAM 2000). The suit is equipped with semiconductor material attached to the performer. Contacts on the fingertips transfer voltage to these sensors. In other systems, such as BodySynth (Van Raalte 1999), sensors detect muscle changes. Sensors attached to the body detect electrical signals generated by muscle contractions. The muscle contractions that trigger sounds can be very subtle, so the same sonic result can be achieved by a wide variety of movements. The MidiDancer system (Coniglio 2000) uses flex sensors and a wireless transmitter to convey joint angle information to a computer. A conceptually similar system based on strain gauges on spring steel and a wireless transmitter is described by Siegel and Jacobsen (1998). Paradiso, Hsiao, and Hu (1999) describe a wireless interface to dancing shoes.

Most of the authors mentioned here also discuss philosophical, practical, and musical implications of sensors for dance and the integration of interactive music with dance performance. We hope that our experience and perspective will add to this growing body of knowledge.

Elements of Integration

In order to achieve a coherent integration of music and dance, we consider their analogous principles. These include exposition of ideas, links, and transitions between sections, variation strategies, embellishment methods, and structural issues.

In dance, exposition of ideas is achieved through the repetition of movements in different space regions, combined with variations, jumps, twists, and falls. Such variations incorporate new movements to those already exposed. The general structure is normally divided into sections, each with its own expressive criteria characterized normally by different scenarios, light, and clothing.

These pose several problems, including what information should be captured, how to characterize this information, and how to relate this information in a way that is coherent with the music. SICIB can be seen as a step toward solving these problems.

Information Captured from the Dancer

All the choreographic information is obtained by SICIB through sensors attached to dancers. A Flock of Birds system (Ascension Technology Corporation 2000) provides six degrees of freedom (three-dimensional position and orientation) tracking at an adjustable sampling rate and connects to a host computer with a serial interface. Each "Bird" is a magnetic tracker that provides up to 144 position and orientation measurements per second within a 10-foot radius around a central transmitter. A balance must be established between the amount of data that needs to be considered in order to capture important choreographic information with sufficient detail and the speed at which such information can be processed. In our experiment, we only consider the position of each sensor in space, and the baud rate of the sensors is fixed at 38,400 bits per second, producing roughly 50 space positions per second per sensor. This sampling rate allows us to create almost continuous changes of musical parameters (e.g., features of granular synthesis, glissando effects, etc.). Real-time considerations and alternative possibilities to capture information by other means are given later in this article.

Choreographic Elements of Control

The data from the sensors are used to obtain the following choreographic information: (1) curvature and torsion of movements, (2) physical position of the dancer, (3) displacement velocity and acceleration, and (4) sudden changes.

One of the fundamental aspects to consider in dance is the curvature and torsion of the movements performed by a dancer regardless of his or her orientation. SICIB uses the Frenet-Serrett theorem, described in the following subsection, to obtain such information.

In general, the physical position of a dancer's ankle or wrist can be used to characterize a particular choreography. For instance, the information of the position of a hand and how it changes through time can have a particular gestural choreographic interpretation. Information about the position of the dancer also allows the segmentation of the choreographic space into regions. Each time a dancer moves into a different region, a new choreographic meaning can be associated with it. We have defined several primitives to easily specify geometrical regions, such as spheres, cylinders, and cubes.

Given a particular time interval, the displacement velocity and acceleration, in each of the space coordinates or its global resultant, can be evaluated to obtain information about a choreography. Finally, sudden changes—including jumps and falls by the dancer—can provide additional choreographic information.

All the above elements are used to characterize different aspects of particular choreographs. Dancers can change any of them with their movements and thus change different aspects of the music. Therefore, we can consider regions that initialize or terminate particular musical events, regions of intensity and pitch, regions associated with particular notes or instruments, regions associated with specific rhythms, etc. Similarly, the velocity and/or acceleration in a particular space direction of a sensor, the curvature and torsion of body movements, and jumps or falls can change any of the musical information. With the space position of the sensors, it is fairly easy to create virtual regions of different geometrical shapes, virtual walls, hallways, and doors (as it will be shown when the experimental results are described).

Frenet-Serrett Theorem

In order to evaluate the curvature and torsion of movements, we are using the Frenet-Serrett theorem (Do Carmo 1976). With it, we can characterize the movements of dancers independently of their orientation. Each sensor reports its position $\{x(t_i), y(t_i), z(t_i)\}$ in space at time t_i. The curvature, κ, of a particular movement is given by

$$\kappa(t_i) = \frac{1}{v(t_i)} \sqrt{\begin{vmatrix} \gamma'(t_i)\cdot\gamma'(t_i) & \gamma'(t_i)\cdot\gamma''(t_i) \\ \gamma'(t_i)\cdot\gamma''(t_i) & \gamma''(t_i)\cdot\gamma''(t_i) \end{vmatrix}} \qquad (1)$$

where $v(t_i)$ is the displacement velocity given by

$$v(t_i) = \sqrt{x'(t_i)^2 + y'(t_i)^2 + z'(t_i)^2}\,. \qquad (2)$$

The expression inside the square root operator is the determinant of the scalar product of γ' and γ'', where

$$\begin{aligned} \gamma'(t_i) &= (x'(t_i), y'(t_i), z'(t_i)) \\ \gamma''(t_i) &= (x''(t_i), y''(t_i), z''(t_i)) \end{aligned} \qquad (3)$$

Here, $x'(t_i)$ represents a change in position or velocity in direction x:

$$x'(t_i) = \frac{x(t_{i+1}) - x(t_i)}{\Delta t} \tag{4}$$

and $x''(t_i)$ represents a change in velocity or acceleration of the x coordinate. The other derivatives of the coordinates y and z are evaluated similarly.

The torsion τ of a particular body movement is obtained from the following formula:

$$\tau'(t_i) = \frac{-1}{(t_i)^2 V(t_i)^6}\left(\gamma'(t_i) \times \gamma''(t_i)\right) \cdot \gamma'''(t_i) \tag{5}$$

where

$$\gamma'''(t_i) = (x'''(t_i), y'''(t_i), z'''(t_i)) \tag{6}$$

and $x'''(t_i)$ represents the change in acceleration in the x coordinate. Since we are using the third derivative of each position (e.g., $x'''(t_i)$), all the calculations can be evaluated from four continuous samples. The user can decide if a simple derivative will be evaluated from two contiguous samples or from samples separated by an arbitrary number of intermediate samples.

Musical Elements

Music can be generated with the aid of Escamol and Aura. In the following sections, a short description of each is given. Interested readers should also consult Dannenberg and Brandt (1996) and Morales-Manzanares (1992).

Escamol

Escamol includes a set of predicates from which music can be generated from simple musical elements, such as pitch, rhythm, and loudness. The musical information that can be affected by Escamol and controlled by the sensors includes the following: duration of a musical event, groups of musical intervals and rhythms associated with the musical event, overall volume of the generated music, musical notes or motives to use as a basis for the generation of musical phrases and/or variations, initialization/termination of predetermined musical events, instrument type (e.g., wind, strings, percussion, etc.) to use in the interpretation of the music generated, and tempo changes of musical sequences that have been generated but not yet interpreted.

Escamol generates music with this information in real time following compositional grammar rules. Such rules determine the criteria (style) for music composition. The rules can be changed by composers to satisfy their compositional preferences and can be changed from one piece to another. With different grammatical rules, Escamol generates music in different styles with the same input information.

Escamol has a library of high-level predicates that can be used to generate music. An example predicate is $\texttt{atasca([N}_1\texttt{/V}_1\texttt{K}_1\texttt{,....., N}_n\texttt{/V}_n\texttt{K}_n\texttt{], Vars, Tempo/ST)}$, where, $\texttt{N}_\texttt{i}\texttt{/V}_\texttt{i}\texttt{K}_\texttt{i}$ refers to the initial note as reference for the voice $\texttt{V}_\texttt{i}$ to be generated in clef $\texttt{K}_\texttt{i}$, `Vars` is the number of variations to consider, and `Tempo/ST` represents the metronome tempo and the starting time *ST*.

The predicate given by `atasca([c4/’4tr’,e5/’5tr’], 5, 60/0)` means that two voices `(c4/’4tr’,e5/’5tr’)` will be generated with five variations, with tempo set to 60 beats per minute and starting immediately at time 0. The starting note of the first voice is C4 (middle C), and it is the fourth voice in treble clef (in case the user wants to generate music notation). Other predicates allow the generation of music from sets of notes, the definition of chords with a variable number of voices, etc.

Escamol can generate more than 20 simultaneous voices, and its predicates include modal counterpoint rules (Fux 1725; Morris 1978), algorithms that simulate Alberti bass and other ostinatti, traditional harmonic progressions, contemporary compositional rules, and generative grammars (Sloboda 1985, 1988; Lerdahl and Jackendoff 1983; Schwananauer and Levitt 1993).

Escamol is written in Prolog (Clocksin and Mellish 1987) with an interface in Tcl/Tk (Welch 1995) and can generate scores in MIDI, Csound (Vercoe 1986), or Aura formats, which allows Escamol to generate music in real time.

Aura

Aura (Dannenberg and Brandt 1996) can be viewed as an object-oriented platform to construct and control software synthesizers. A goal of Aura is to provide flexible, real-time, low-latency sound synthesis, so Aura was a natural choice for this work. Objects in Aura include musical instruments, and their attributes are used to specify pitch, attack, envelopes, vibratos, loudness, duration, etc. Aura offers great flexibility to the user, as it allows a musician to configure new instruments and control strategies. Objects send and receive information using timed asynchronous messages, which supports real-time execution.

Sound is generated in Aura by creating and "patching" together various objects. For example, an oscillator can be created and patched to the Audio Output object to play a tone. Messages of the form "Set *attribute* to *value*" are used to control parameters such as frequency and amplitude. Aura offers a large number of built-in sound objects, including oscillators, filters, envelope generators, sample players, and mixers. New objects can be written in C++.

Aura also includes a simple scripting language allowing users or other programs to instantiate objects, patch them together, and set parameters in real time using text-based commands. This text interface, combined with Unix pipes, allows Aura to communicate with SICIB and Escamol.

Integration Between Music and Dance

The choreographic elements previously described can affect any of the musical elements described in the previous subsection. The association between dance and music is specified by a rule-based language that allows one to define which choreographic aspects to consider from a particular sensor, which musical aspects are controlled by that sensor (i.e., what the meaning is, in musical terms, of the choreographic changes of a sensor), and which musical and choreographic aspects (rules) are given preference over other ones.

The information from the sensors is transmitted with the predicate `pos(sensor_ID, (X,Y,Z), Time)`, where `sensor_ID` is a unique name associated with each sensor; `(X,Y,Z)` are its space coordinates; and `Time` is a time tag indicating when the data was produced.

Choreographic regions can be specified. A rectangular region is specified by two spatial points (the lower left-hand side corner and the upper right-hand side corner) as `re_region(region_ID (X1, Y1, Z1),(X2, Y2, Z2))`, where `region_ID` is a unique name associated with each region. Other regions can be specified as well. For instance, spherical and cylindrical regions can be defined with an inner and an outer radius, considering the center of the region to be the reference point of the Flock of Birds transmitter:

sp_region(ID$_{region}$, radius$_1$, radius$_2$)
cy_region(ID$_{region}$, radius$_1$, radius$_2$)

Here, we assume that a cylindrical region has an infinite height. An entire sphere or cylinder would have the inner radius set to (0,0,0). The definition of regions allows one to define virtual walls (thin regions) and create different scenarios. A user can define as many regions as desired and use them in the rules. Rules in SICIB have the following syntax:

S$_i$ IF <condition>*
THEN <action>*

where S$_i$ is the name of the *i*th sensor, `<condition>*` is a conjunction of choreographic conditions that must be satisfied, and `<action>*` is a conjunctive set of musical actions to perform.

The conditions can take one of several forms. The construct `curve_&_torsion(S,T,DT,K,R)` evaluates the curvature `K` and torsion `R` followed by a sensor `S` starting from cycle `T` and considering measurements back in time every `DT` cycles. The construct `pos(S,P,T)` tests if the position of a sensor is within a specific region, and `Jump(S,T,H)` tests for a change above a particular height `H` of sensor `S` at cycle `T`. Finally, `fall(S,T,H)` tests for a change below a particular height `H` of sensor `S` at cycle `T`. Arbitrary Prolog predicates are also possible.

Other predicates for comparing positions of sensors have been defined but were never used in the experiments. In the above conditions, if a cycle T is not specified, the most recent cycle is considered (i.e., the most recent data from the sensors). This parameter allows one to specify delayed reactions to particular choreographic events.

The allowed actions include arbitrary Escamol predicates, arbitrary Aura commands, and arbitrary Prolog predicates.

The rules can change graphical displays as well. Although a rudimentary interaction has been implemented, any of the choreographic elements previously described can change graphical displays, such as color, rotation, and/or displacement of graphical figures, etc., using any of the capabilities of Performer (a graphical tool that runs on Silicon Graphics workstations) through Escamol predicates.

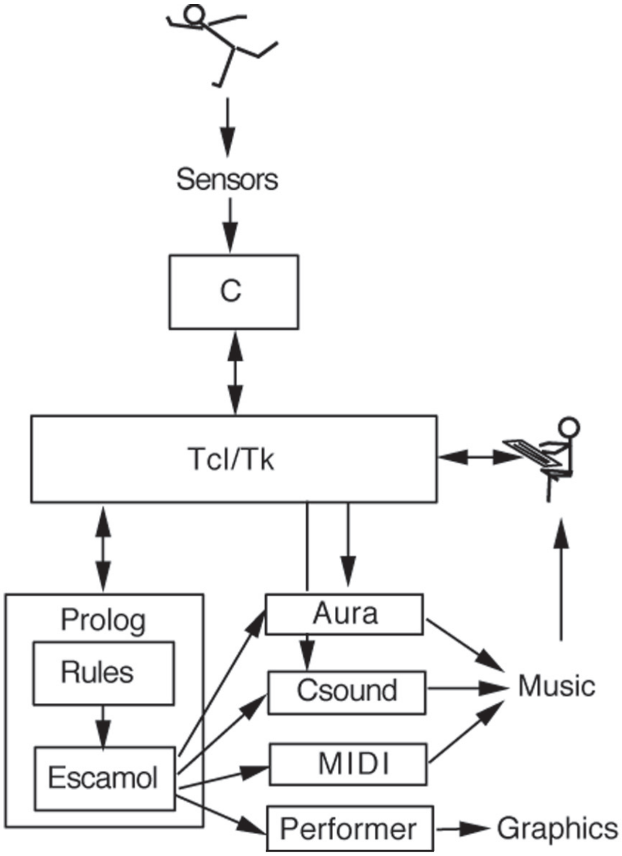
Based on information from the sensors, SICIB evaluates which rules should be followed given the choreographic information considered in its current set of rules. All the rules that satisfy the current set of conditions can be active at the same time, which allows one to mix several choreographic elements simultaneously. Alternatively, the first rule (in the set of rules) that satisfies its conditions can be executed. This option imposes a rule order, thereby allowing one to specify precedence criteria among rules (i.e., the first rules have higher precedence than the later rules).

The Architecture of SICIB

The control of SICIB is driven by an interface written in Tcl/Tk, as shown in Figure 1. With the interface, a user can start Prolog (to start the integration rules and Escamol), Csound, or Aura, and a sensor data program written in C. The sensor data program sets the serial port of a Silicon Graphics workstation, specifies the number of sensors to use, their baud rate (currently fixed at 38,400 bits per second), and the desired information (position in space). The information from the sensors is decoded and scaled down by a constant factor for convenience. The C program also filters the information produced by the sensors and adds a time tag to them. Positions that do not change by more than a particular threshold (which could vary for each sensor) are ignored, which means that musical changes can only be produced with body movements. Despite the robustness of the sensors, the filter is also used to eliminate the maximum and minimum values of five sequential samples when the curvature and torsion predicate was evaluated (because with second and third derivatives the results are very sensitive to noise).

The filtered data is fed to a Tcl/Tk function through a Unix pipe and sent to a Prolog program. The Prolog program includes Escamol and the "if-then" rules described in the previous section. From the data of the sensors, different choreographic information is obtained, and the condi-

Figure 1. The control of SICIB is driven by an interface written in Tcl/Tk.



tions of the rules are tested. The rules whose conditions are satisfied are activated and their musical actions performed. Musical actions invoke Escamol to generate music, which is then sent to Aura. Alternatively, musical actions can be sent directly to Aura.

In SICIB, different rule files can be used for different choreographic pieces—even during the same piece. This means that different musical and choreographic meaning can be attributed to the same sensor at different instances in time (within a choreography or between choreographs). The user and the dancer can change at any time the rule set through SICIB's interface or through body movements, respectively. Additionally, the information from the sensors can be activated or deactivated at different times during a particular performance through the interface.

Experiments and Results

Several live performances have employed SICIB. In this section, three pieces of the most recent performance are described, illustrating some of SICIB's capabilities. All of them were performed by one dancer and live musicians (playing flute, clarinet, and piano). In all three pieces, the dancer using SICIB generated the only electroacoustic music. Three different composers—Jonathan Berger (Stanford University), Roger Dannenberg (Carnegie Mellon University), and Roberto Morales (University of Guanajuato, Mexico)—worked with a choreographer and dancer (Raúl Parrau) in the definition of their pieces. The three pieces were put together for a single concert by the composers and the dancer/choreographer. For practical and aesthetic reasons, they use similar choreographic features.

The longest delays in the three pieces between motion and the resulting sound were close to 0.5 sec. The sensors were attached to the dancer and suspended from the ceiling, which imposed movement restrictions on the dancer and demanded creative solutions. The dancer felt constrained at the beginning of the experiments but was able to design adequate choreographs and was not affected in his performance.

Jonathan Berger's *Arroyo*

Arroyo is part of a set of concert works and sound installations in which a virtual labyrinth is created using sound to represent physical delimiters such as walls or ceilings. In *Arroyo*, the dancer must traverse a three-dimensional virtual maze, relying solely upon sonic cues as a guide. The piece thus considers virtual reality from a sensory-deprived standpoint rather than providing the standard barrage of multi-sensory stimuli. As in any maze, backtracking and retracing steps provide the only way out of wrong turns and mistaken paths. Thus, musical structure is provided by this natural process of "finding the way." (In the future, a number of mazes will be provided by the composer for the piece. An interface in which performers or listeners can create their own mazes will also be added.) The instrumentalists react to the dancer's arrival at key points in the maze by augmenting the digital audio cues and clues; they also provide occasional "hints" when the dancer seems to be floundering.

The maze in *Arroyo* is taken from an archaeological map of part of the Chaco de Arroyo, an 11th century Anasazi edifice. The maze is comprised of nine chambers (see Figure 2). Each room corresponds to a musical segment. The dancer wears three sensors: one sensor on each wrist and one sensor on an ankle. The sensors of the hands are used to detect the virtual walls of the maze. Each time one of these sensors "touches" a virtual wall (i.e., crosses a region), a particular sound is produced. These sensors are used by the dancer as feedback to identify the walls and entrances of the maze. The dancer's goal is to reach the central room. The sensor on the ankle detects the room (region) where the dancer is placed. Each time the dancer moves to a different room (region), a unique musical event associated with that particular room is triggered. The electronic sounds are derived from a digitally processed squawk of a macaw (a South American bird whose skeletal remains were found in the central room of the Arroyo site). As the dancer enters a room, the live musicians segue into the musical material that corresponds to that room.

Figure 2. *The maze in* Arroyo, *comprised of nine chambers.*

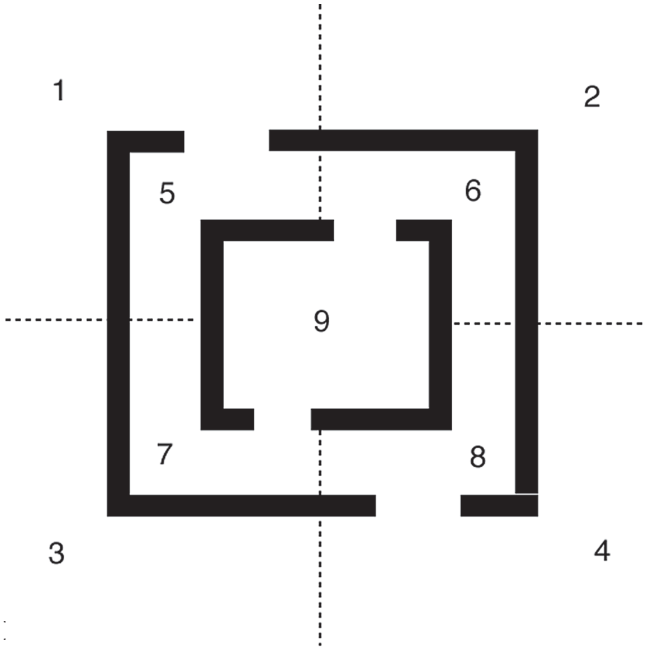
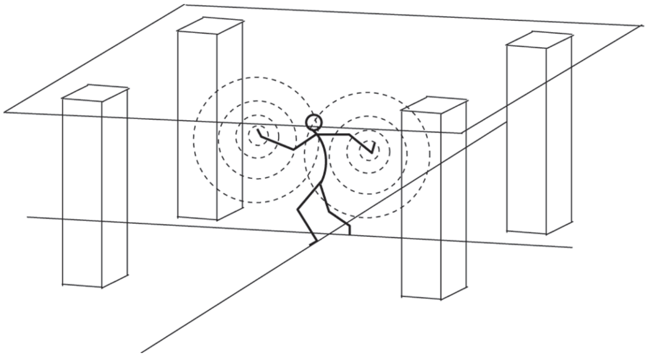


Figure 3. *A representation of the region and height rule sets for SICIB.*



Roger Dannenberg's composition *Aura* involves two sensors that are attached to the wrists of a dancer. There are several rule sets for the sensors (i.e., the sensors have different musical and choreographic meaning at different times during the piece). The rule sets are changed using SICIB's graphical interface as scored music is performed. One of the rule sets defines four rectangular columns (regions) and a region above a certain height (see Figure 3). Each time a dancer "touches" one of the rectangular regions, Escamol generates a musical phrase that is synthesized by Aura. Similarly, if the dancer raises an arm above a particular height, a musical event is produced but this time only once (i.e., he/she needs to lower his or her arm and raise it again to produce another musical event). Other rule sets define an environment where the movements of the dancer change different parameters of granular synthesis. The position of the sensors allows an almost continuous change in some continuous musical parameters used for granular synthesis. Another rule set is used to gradually lower the volume of the music, producing more natural transitions between rule sets. In addition, a projected computer animation is changed each time a musical event is triggered.

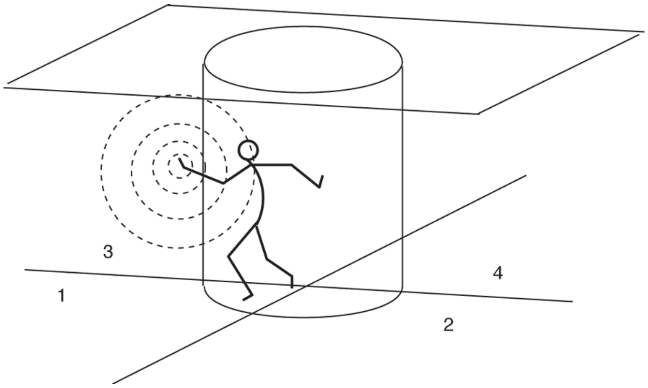
Roberto Morales-Manzanares' *Trío de Cuatro*

Roberto Morales-Manzanares' composition *Trío de Cuatro* also uses two sensors attached to the wrists of a dancer. The piece includes two different sets of rules that are controlled by the dancer. One set divides the space into four regions, each with a different granular synthesis motive. The dancer can change regions and continuously alter granular synthesis parameters with his or her movements. The other rule set defines a central cylindrical region with a surrounding wall. Inside the region, no sound is produced. Each time the dancer "touches" the wall with either hand, a sound is produced. Outside the region, notes are produced according to the distance from the center, creating a glissando effect with the dancer's movements. The closer the dancer is to the cylinder, the higher the pitch of the notes produced. This allows the dancer to create musical dialogues with a live musician. Additionally, the dancer can toggle between rule sets by raising a hand above a certain height. The height value can be set sufficiently high so that the dancer can only change rule sets with a jump, creating a more dramatic effect (see Figure 4).

Experiences with SICIB

The development of SICIB provided several insights regarding what to consider in the development of similar systems. The most difficult and time-consuming part was programming the communication with the sensors and calibrating them.

Figure 4. To change rule sets, the dancer simply moves beyond the defined region.



Michael Lee's contribution in this area was crucial for the success of SICIB. Once the communication program was properly running, a much more creative environment was established for composers and choreographers.

Another critical aspect to consider in systems like SICIB is possible time delays that can occur between the dancer and the music produced by the system as a consequence of the movements. The audience must feel the interaction of the dancer with the music, and long time delays cannot occur. Because the sensors can produce a large amount of data every second, data reduction is essential, and several considerations emerged from practice and experimentation. Rules that can be activated very often (e.g., with any movement of the dancer) are normally related either with short musical events or changes in musical parameters. If a composer wants to trigger long musical sequences, either those rules are activated occasionally (owing to the nature of the particular choreographic event considered for it), or they are activated only once (until another equivalent choreographic event is encountered).

In some experiments, where we allowed musical events to be present and generated simultaneously, it was difficult to differentiate what the sensors were doing, and the system could eventually be saturated with musical output. There must be a balance between the number of notes (related with the length in time of the generated music) and the rate at which the movements (and therefore music) are performed. A close coordination with the dancer can avoid these problems. Slow movements can still elicit a rapid series of musical events, and fast movements can control long sustained sounds.

It is important for dancers to learn the musical consequences of their movements. We have used at most four sensors on a dancer, as the dancer can be overloaded with information, making it difficult to follow the musical implications of physical gesture.

It is important to have adequate time and space for testing and rehearsing. One of the most artistically limiting factors for us has been the difficulty of assembling composers, dancer, and equipment in a large dance space with enough time to experiment and rehearse.

Once the sensors were properly running, the rich choreographic vocabulary, Escamol's and Aura's music capabilities, and the natural coupling between them through a simple rule-based system, allowed the composers to express their rules in relatively short times. We spent less than one day on the definition and tuning for each of the above pieces.

Even though the pieces do not include all the subtleties of the choreographic and music capabilities described in the paper, these features are available in SICIB and were tested in simple experiments. For practical and aesthetic reasons, they were not included in the previously described pieces.

Conclusions and Future Work

With SICIB, a dancer need not adjust movements to a predefined musical piece. This allows greater freedom and opens new areas for dance improvisation and performance. From a musical perspective, SICIB represents a new virtual instrument that produces music through body movements, offering new possibilities for music composition, improvisation, and performance. SICIB has been primarily used as a dialogue facilitator between live performance musicians and dancers, producing sounds with their movements. We believe that this interaction has been made possible by SICIB's flexibility.

The use of a very high-level language (Prolog) greatly facilitated making adjustments in the limited rehearsal times available to us on stage. In principle, we could have done everything directly in C++ using Aura, but delegating user interface, composition, and control tasks to other programs made more sense and worked out quite well. This experience has prompted the design of a very high level embedded language for the next version of Aura.

Although several systems have been proposed with very good results, we believe that SICIB offers a good alternative with rich choreographic information (such as curvature and torsion of movements, falls, jumps, acceleration, continuous changes, regions, etc.), complex musical events, and a combination of continuous parametric control and discrete "triggers" of musical events within a single, flexible control scheme. Finally, the syntax employed in the rules to link music and dance is fairly simple yet quite powerful.

There are several future research directions that we are considering at the moment. In particular, we need a broader engagement with the use of SICIB by dancers and musicians. We want the dancer not to think too much about his or her movements, yet produce a coherent piece of music. At the same time, as the music produced by the dancer may be just a part of a larger ensemble, we want coherence in the entire piece. This may require sensors for live musicians as well as live dancers.

Although the choreographic primitives in the rules have been adequate so far, we would like to explore new primitives and provide the user with a three-dimensional graphical interface to define regions. We must experiment with several dancers at the same time and with wireless sensors. We are also exploring the use of cameras to capture the whole body position. Finally, we are also considering ways to correlate lighting effects with a dancer's movements.

Acknowledgments

We would like to thank the group of Centro Multimedia (CM) from Centro Nacional de las Artes (CNA), especially to director Andrea Di Castro for his continuous support in the realization of this work. We also thank Fideicomiso para la Cultura México—Estados Unidos and Raúl Parrao for his valuable suggestions and astonishing performances, Michael Lee who contributed significantly to the development of SICIB, and Silicon Graphics and Ascension Technologies who kindly loaned equipment for performances.

References

Allen, J. 1987. N*atural Language Understanding*. Menlo Park, California: Benjamin/Cummings.

Ascension Technology Corporation. 2000. "Ascension Technology Corporation Flock of Birds." http://www.ascension-tech.com/products/flockofbirds/flockofbirds.htm.

Camurri, A., et al. 2000. "EyesWeb: Toward Gesture and Affect Recognition in Interactive Dance and Music Systems." Computer Music Journal 24(1):57–69.

Clocksin, W. F., and C. S. Mellish. 1987. *Programming in Prolog.* Berlin: Springer-Verlag.

Coniglio, M. 2000. "MidiDancer." http://www.troikaranch.org/mididancer.html.

Dannenberg, R., and E. Brandt. 1996. "A Flexible Real-Time Software Synthesis System." *Proceedings of the 1996 International Computer Music Conference.* San Francisco: International Computer Music Association, pp. 270–273.

Do Carmo, M. P. 1976. *Differential Geometry of Curves and Surfaces.* Upper Saddle River, New Jersey: Prentice Hall.

Fux, J. J. 1725. *Gradus ad Parnassum*, trans. Alfred Mann. New York: Norton, 1943.

Griffith, N., and M. Fernstrom. 1998. "LiteFoot—A Floor Space for Recording Dance and Controlling Media." *Proceedings of the 1998 International Computer Music Conference.* San Francisco: International Computer Music Association, pp. 475–481.

Johnstone, E. 1991. "A MIDI Foot Controller—The PodoBoard." *Proceedings of the 1991 International Computer Music Conference*. San Francisco: International Computer Music Association, pp. 123–126.

Lerdahl, F., and R. Jackendoff. 1983. A Generative Theory of Tonal Music. Cambridge, Massachusetts: MIT Press.

Maes, P., et al. 1995. "The ALIVE System: Full-Body Interaction with Autonomous Agents." *Proceedings of the Computer Animation '95 Conference.* Piscataway, New Jersey: IEEE Press, pp. 11–18.

Morales-Manzanares, R. 1992. "Non Deterministic Automatons Controlled by Rules for Composition." *Pro-*

ceedings of the 1992 International Computer Music Conference. San Francisco: International Computer Music Association, pp. 400–401.

Morales-Manzanares, R., and E. Morales. 1997. "Music Composition, Improvisation, and Performance Through Body Movements." Paper presented at the AIMI International Workshop KANSEI: The Technology of Emotions, 3–4 October, Genova, Italy.

Morris, R. O. 1978. *Contrapuntal Technique in the Sixteenth Century.* Oxford: Oxford University Press.

NoTAM. 2000. "Control Suit." http://www.notam.uio.no/notam/kontrolldress-e.html.

Paradiso, J., et al. 1997. "The Magic Carpet: Physical Sensing for Immersive Environments." Proceedings of the International Con*ference on Computer Human Interfaces*. New York: Association for Computer Machinery, pp. 277–278.

Paradiso, J., K. Hsiao, and E. Hu. 1999. "Interactive Music for Instrumented Dancing Shoes." *Proceedings of the 1999 International Computer Music Conference*. San Francisco: International Computer Music Association, pp. 453–456.

Pinkston, R., J. Kerkhoff, and M. McQuilken. 1995. "A Touch Sensitive Dance Floor/MIDI Controller." *Proceedings of the 1995 International Computer Music Conference*. San Francisco: International Computer Music Association, pp. 224–225.

Rokeby, D. 1999. "The Very Nervous System." http://www.interlog.com/~drokeby/vns.html.

Schwananauer, S., and D. Levitt. 1993. *Machine Models of Music*. Cambridge, Massachusetts: MIT Press.

Siegel, W., and J. Jacobsen. 1998. "The Challenges of Interactive Dance: An Overview and Case Study." *Computer Music Journal* 22(4):29–43.

Sloboda, J. 1985. *The Musical Mind: The Cognitive Psychology of Music.* Oxford: Clarendon Press.

Sloboda, J., ed. 1988. *Generative Processes in Music.* Oxford: Clarendon Press.

STEIM. 2000. "BigEye." http://www.steim.nl/bigeye.html.

Van Raalte, C. 1999. "Old Dog, New Tricks. Biofeedback as Assistive Technology." Paper presented at the California State University, Northridge Center On Disabilities 1998 Conference, March 1998, Los Angeles, California. Also available online at http://www.dinf.org/csun_98/csun98_098.htm.

Vercoe, B. 1986. "Csound: A Manual for Audio Processing System and Supporting Programs." Program Documentation. Cambridge, Massachusetts: MIT Media Lab.

Welch, B. 1995. *Practical Programming in Tcl and Tk*. Englewood Cliffs, New Jersey: Prentice Hall.

Winkler, T. 1995. "Making Motion Musical: Gesture Mapping Strategies for Interactive Computer Music." *Proceedings of the 1995 International Computer Music Conference*. San Francisco: International Computer Music Association, pp. 261–264.