Gestural Control of Music Using the Vicon 8 Motion Capture System

Christopher Dobrian
University of California, Irvine
303 Music & Media Bldg., UCI
Irvine, CA 92697-2775
(1) 949-824-7288
dobrian@uci.edu

Frédéric Bevilacqua
University of California, Irvine
1002 Health Sciences Road East
Irvine, CA 92612-1475
(1) 949-824-4104
fbevilac@uci.edu

ABSTRACT
This article reports on a project that uses unfettered gestural motion for expressive musical purposes. The project involves the development of, and experimentation with, software to receive data from a Vicon motion capture system, and to translate and map that data into data for the control of music and other media such as lighting. In addition to the commercially standard MIDI—which allows direct control of external synthesizers, processors, and other devices—other mappings are used for direct software control of digital audio and video. This report describes the design and implementation of the software, discusses specific experiments performed with it, and evaluates its application in terms of aesthetic pros and cons.

Keywords
Motion capture, gestural control, mapping.

1. INTRODUCTION
The Vicon 8 motion capture system[1] is recognized to be one of the best available systems for accurately recording three-dimensional movement, particularly—but by no means exclusively—movement of the human body. At the University of California, Irvine (UCI), a small group of artists and programmers is working in UCI's Motion Capture Studio, in conjunction with the Realtime Experimental Audio Laboratory (REALab), to develop software for the translation and mapping of Vicon motion capture data—which is normally used for animation or for biomechanics research—into control data for music, as well as for other media such as lighting, digital audio, and digital video. The working name of this software is MCM (Motion Capture Music).

The team is developing MCM concurrently on two programming platforms. One group is coding it in generic Java and C++ for optimal portability to any operating system, translating received motion capture data into directly transmitted MIDI data. A second group is coding extended functionality in Max[2]. The Max version of MCM is presently limited to the Macintosh operating system, but receives the realtime motion capture data via Open Transport UDP. The advantage of using Max is that it already provides comprehensive capabilities for the control of digital audio and video—via MSP and Jitter—and therefore allows direct mapping of motion capture data to those media as well as to MIDI.

The goal of the software development is to create a straightforward, useful tool for employing gestural motion to control audio-visual performance media with accuracy and reliability. Members of the team are conducting experiments with this type of control, and these experiments are guiding the software development as it is in progress. What follows is a report on this work.

2. DATA FROM THE VICON V8 SYSTEM
The details of the Vicon system are adequately described elsewhere[1,3], so here we discuss only the data it produces. The data derived from the captured motion are most commonly saved to disk, as a Vicon-standardized .C3D file. Captured data files are then normally used as input to an animation program such as 3D Studio Max for realistic generation of lifelike animated characters, or used for biomechanical studies of bodily motion (sports, physical therapy, ergonomics, etc.). In the .C3D format, each frame of information is represented as a list consisting of Cartesian *x, y, z* coordinates in 3D space for each marker. The Vicon 8 system at UCI reports up to 120 frames per second; more modern Vicon systems boast a much higher frame rate. The user determines the ordering of the markers in the list when recording the data.

2.1 Vicon Real-time Data
The Vicon system has recently become of interest for musical expression because of the availability of the Vicon Real-time Engine[4], which allows full realtime transmission of the motion capture data. For this project, Vicon Motion Systems provided their in-house RTEmulator software, which allows one to emulate the behavior of the Real-time Engine. RTEmulator reads data from a .C3D file and transmits it in the format of the Real-time Engine, allowing simulation and testing of realtime motion capture without having the Vicon Real-time Engine itself. Realtime access to Vicon data makes the system useful as a tool for musical expression.

3. INITIAL DESIGN OF MCM
The initial design[5] for MCM intended to make as simple—and as simple to use—a program as possible for mapping motion capture data to musical control data. The design allows for the user to select a marker (i.e. a position on the body), a coordinate (*x*, *y*, or *z*), and a range of space to in which to track that coordinate, and linearly map that data to any range of MIDI values for any MIDI channel message. The user can specify as many such mappings as desired. The intention was to make the most direct possible way for any motion capture parameter to be used to control a MIDI device. For more complex mappings, it was assumed that an additional program would mediate the transmission between MCM and the MIDI device(s).

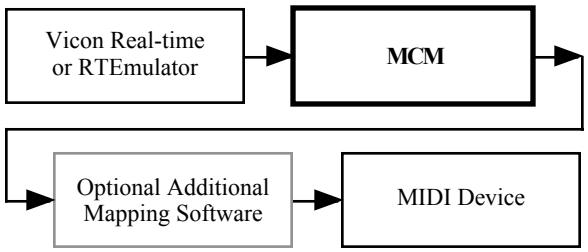

Figure 1. Overview of Generic Usage of MCM

3.1 Specifying Mappings and Translating Data

Per this initial design plan, the implementation of MCM consists of two parts: a user interface for specifying the desired mappings of marker coordinates to MIDI data, and a translator that uses those specifications to perform the desired mappings on incoming motion capture data and transmit the appropriate MIDI data. These two parts are implemented as separate programs, called MCMMap and MCMTranslate.

3.1.1 MCMMap

MCMMap provides the user with panels (as many as desired) for inputting the desired mappings. The parameters to be specified by the user are marker, coordinate, coordinate range minimum and maximum, MIDI message type, MIDI range minimum and maximum, and MIDI channel. Depending on the type of MIDI message selected, other parameters may appear: e.g., controller number if a control change message is selected, velocity and duration if a note message is selected, etc. A session of such specifications is saved in a plain text file with a .mcm tag. The .mcm file is used by MCMTranslate to map motion capture data to MIDI data.

3.1.2 MCMTranslate

MCMTranslate performs four tasks: it receives motion capture data from Vicon Real-time or RTEmulator, uses a .mcm file of mapping specifications to translate specific input values into MIDI messages, transmits the MIDI messages, and schedules note-off messages in the future to end any note-on messages it has transmitted. Since all of the mapping specifications are made in MCMMap, MCMTranslate requires nothing more of the user than to start it and select a .mcm file for the translation of the incoming data.

4. IMPLEMENTATION IN JAVA/C++

A relatively "platform-neutral" version of MCM is being implemented according to the initial design described above, using Java for MCMMap and C++ for MCMTranslate. The intention is for it to be maximally compatible with the operating system used by the Vicon system itself, Windows NT/2000, yet written with minimal reliance on OS-specific functionality, so as to be as easily portable as possible to any new OS. This implementation was begun by undergraduate students from the UCI Department of Information and Computer Science (ICS), Cayci Suitt and Gene Wie, and is being completed by ICS students Mark Magpayo and Maybelle Tan.

5. IMPLEMENTATION IN MAX

The authors are also implementing an extended version of MCM in the Max programming environment. The Max version (MCMMax) incorporates a number of extensions to the basic design that make it more versatile and useful for more complex mappings of gesture to music. Because Max is an inherently object-based "patchable" system of modules, MCMMax allows for easy redirection of input data to different types of mapping. Because Max already has considerable features for audio and video—MSP and Jitter—incorporated directly in the programming environment, it's a simple matter to create new modules for controlling these media as needed; the mapped data can thus control MIDI synthesizers and sound processors, digital audio, and digital video all within the same software system.

5.1 Design Extensions in MCMMax

MCMMax extends the initial design by permitting tracking of more types of input information, selection from a variety of different mapping schemes, and selection of different destinations for the mapped data.

5.1.1 Information Derived from the Received Data

The motion capture data received from the Vicon system consists of *x, y, z* position values for each marker being tracked. In addition to the position of any marker in any dimension, MCMMax allows the user to track other information: marker velocity in one, two, or three dimensions; marker acceleration in one, two, or three dimensions; the distance between any two markers; and the angle formed by any two or three markers. As with the basic version of MCM, one can specify a range of input values to track.

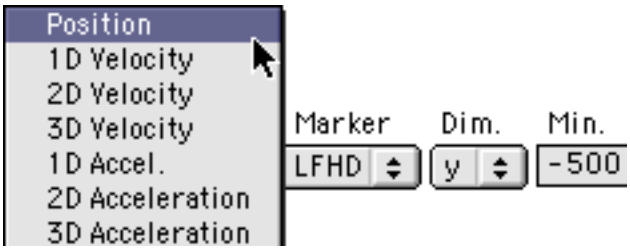

Figure 2. Input Selection in MCMMax

5.1.2 Mappings

In addition to the linear mapping used in MCM, MCMMax also provides the choice of reversed, exponential, logarithmic, and non-linear (lookup table) mappings. When the user selects exponential or logarithmic mappings, a new field appears for entering an exponent or base; when the user selects non-linear mapping, the user may open a stored lookup table or draw an arbitrary mapping curve.

5.1.3 Destinations for Mapped Data

In MCMMax, transmitting mapped data as MIDI messages is just one of the possible choices of destination. Because of the ease with which the data can be routed to different subroutines in Max, the data can just as easily be mapped to serve as control data for digital audio in MSP or digital video in Jitter. We have already performed experiments with mapping motion capture data to MSP parameters (e.g., filter cutoff frequency, etc.), and will continue to develop a variety of different mapping modules that can easily be plugged into MCMMax as new features.

5.1.4 Gesture Detection and Recognition

Some of the research work in the UCI Motion Capture Studio has focused on the detection of individual gestures within the stream of motion capture data, and on recognizing particular kinds of gestures.[3] Such gesture detection has been demonstrated to be feasible and musically useful, and the results of those experiments—edge detection in acceleration curves to detect important changes of gesture (by ICS graduate

student Jeff Ridenour), and principle component analysis to recognize particular types of gesture (F. Bevilacqua)—may also be easily encapsulated as plug-ins for MCMMax.

6. AESTHETIC DIRECTIONS

The numerical and musical problems of mapping captured gestural data to musical control have been discussed in writings by these authors and others.[3,6,7] We will point out here the primary challenges in the development of new musical performances using the Vicon Motion System as input for realtime gestural control of music.

6.1.1 Performing the Virtual Instrument

A performer of this system directly produces and controls musical events with no tangible physical interface, and is thus performing a purely "virtual" instrument. There is no haptic feedback as there is with any physical device, and no precedent or restriction guiding or dictating the gesture. These facts can be viewed as challenges or obstacles to performance of deterministic composed music, but can be seen as opportunities for improvisation and for the discovery of "musicality" inherent in bodily movement. We are thus interested more in the immanent musicality of the human form than in a proof of concept of yet another alternative controller. This pursuit requires that performer(s) be skilled in both musical improvisation and movement, yet not be tied to traditional dance or music vocabularies. That is a significant aesthetic challenge, but one that encourages and requires the collaboration of composers, programmers, and performers.

6.1.2 Multiplicity of Parameters

A single performer wearing a standard set of thirty markers, with three coordinates per marker, produces a stream of 90 simultaneous continuous parameters available for musical control. (To say nothing of other available information such as velocity, acceleration, distance, angle, etc., or of multiple performers.) This profusion of control data presents a management challenge for the composer, and a challenge of the limitations of awareness for the performer. To solve this problem by using only a small number of marker coordinates would be to ignore the unique power and potential of this system. The opportunity and challenge of this system is to devise strategies for mapping so very many degrees of freedom into a meaningfully expressive whole.

6.1.3 Lack of Portability

The Vicon 8 system requires circular placement of its eight cameras, and takes some time to set up and calibrate, so it is not well suited to a traditional proscenium concert situation. The fact that the Vicon Real-time Engine is a TCP/IP server, however, means that the animation software, MCM, and any other performance components need not be in the same location as the Vicon 8 system. The performer may be in a remote location, and be seen by the audience only in the form of an animated, musical avatar. This presents opportunity for new formats of performance and interactivity.

7. FUTURE USES

The MCM project provides easy mapping of motion capture data to musical control. This will give Vicon users in the field of animation the ability to experiment with musical soundtracks ideas that are directly generated by the same data as is driving the animation. MCM provides researchers working on problems of gesture detection and recognition in the UCI Motion Capture Studio with a modular set of mapping tools into which they can interject motion analysis components. MCM makes the Vicon system into an instrument for musical expression, and gives the composer access to a remarkable affluence of simultaneous control data all coming from the bodily motion of a single performer with no physical interface.

8. ACKNOWLEDGMENTS

This project has been supported by the facilities of the Realtime Experimental Audio Laboratory and the Motion Capture Studio at the University of California, Irvine. Vicon Motion Systems has provided us with their RTemulator software to assist us in this software development. We acknowledge the contribution of Lisa Naugle to the original design, the work of programmers Cayci Suitt and Gene Wie in writing the design specifications with valuable advice and supervision by Andre van der Hoek, Ms. Suitt for programming the original mapping interface, and programmers Maybelle Tan and Mark Magpayo for their work in completing the mapping and scheduling software for PC.

9. REFERENCES

[1] Vicon 8 motion capture system, http://www.vicon.com/entertainment/technology/v8.html

[2] Max/MSP programming environment. http://www.cycling74.com/products/maxmsp.html

[3] Bevilacqua, F., J.Ridenour, and D. Cuccia, "Mapping Music to Gesture: A study using 3D motion capture data", Proceedings of the Workshop/Symposium on Sensing and Input for Media-centric Systems, Santa Barbara CA, 2002.

[4] Vicon Real-time motion capture system, http://www.vicon.com/main/technology/realtime.html

[5] Suitt, C. and G. Wie, MCM Design Document, http://www.solscope.com/~gwie/ics125/ mcmdes06052001.pdf

[6] Dobrian, C., "Aesthetic Considerations in the Use of 'Virtual' Music Instruments" Proceedings of the Workshop on Current Research Directions in Computer Music, Institut Universitari de l'Audiovisual, Universitat Pompeu Fabra, Barcelona, Spain, 2001.

[7] Wanderley, M. and M. Battier, eds., Trends in Gestural Control of Music, Paris: IRCAM - Centre Pompidou, 2000.

[1] A demonstration of MCM, including audio-visual examples and explanations, is available online at http://music.arts.uci.edu/dobrian/motioncapture/