THE M-OBJECTS: A SMALL LIBRARY FOR MUSICAL RHYTHM GENERATION AND MUSICAL TEMPO CONTROL FROM DANCE MOVEMENT IN REAL TIME

Carlos Guedes

New York University
Escola Superior de Música e das Artes do Espectáculo
Instituto Politécnico do Porto, Portugal
e-mail: carlos.guedes@nyu.edu

ABSTRACT

This software library allows a dancer to control the tempo of an electronically-generated music score, and/or to generate musical rhythmic structures from bodily movement in real time in interactive dance performance. The movement data is gathered non-invasively, using a fixed video camera placed outside the area in which the dancer is moving. This library is implemented as a set of external objects, called the m-objects, for the modular programming environment Max/MSP [1] using an external object or library that performs frame-differencing analysis of a video stream in real time. This library should also be easily ported to other visual programming environments that perform video analysis in real time such as Isadora [2] or EyesWeb [3]. During the performance of a piece utilizing this software, the musician can give or take away the control over rhythm generation and/or musical tempo from the dancer, and substantially alter the musical content of a piece during performance. In this paper I describe each object comprising the library. In the demonstration session I will show the basic connections that have to be performed among the objects in order to generate musical rhythm and to control musical tempo from dance movement.

1. INTRODUCTION

The m-objects are small library of Max/MSP externals designed for interactive dance performance. These objects allow the creation of a *musical channel of communication* between dancers and musicians in computer-mediated collaborations in real time. The m-objects perform certain types of analysis and processing of dance movement as captured by a video camera, and extract what I call *musical cues* from dance movement in real time. *Musical cues* are rhythms produced by movement in dance that bear qualities akin to musical rhythms in their durational and articulatory nature. As noted by Fraisse [4] [5], and Parncutt [6], the periodicities found in musical rhythms lie between 200 and 1800 ms. This range is also the one that allows for motor synchronization with a sound stimulus. The *musical cues* extrapolated by the software are thus the articulations in dance movement whose time spans fall within that range.

During an interactive dance performance utilizing the m-objects, a musician can utilize the *musical cues* extracted from dance movement to either generate rhythm to animate a musical sequence or to enable the dancer the control of the musical tempo of an electronic music score in real time.

2. CONCEPTUAL FRAMEWORK FOR SOFTWARE DESIGN

Schematically, the software that was developed works as shown in Figure 1.
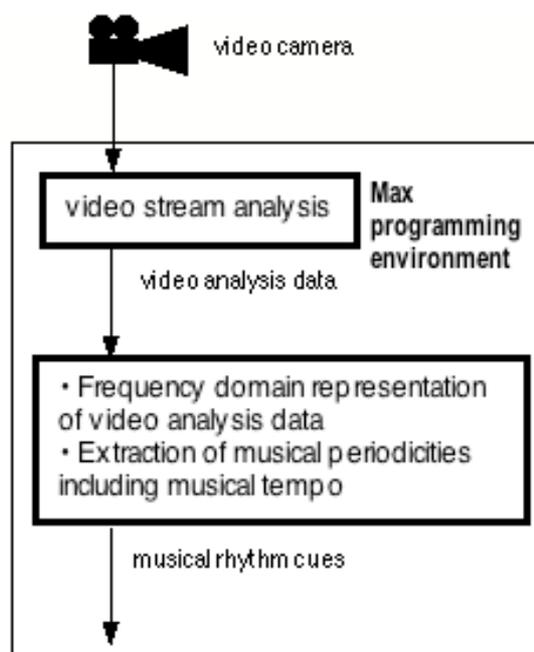
Figure 1. Schematic representation of the software.

A fixed video camera grabs the movement data at 25 or 30 frames per second. The video signal is digitized and the sum of pixels that changed brightness between consecutive frames are computed (a technique commonly known frame differencing). This can be done using

Cyclops [7] or SoftVNS 2 **v.motion** [1] object [8]. Subsequently, that time-varying data is given a representation in the frequency domain, and the *musical cues* are extracted from dance movement in real-time. The user of the system can map those data directly, for example, by utilizing the frequencies extracted from dance movement that have musical relevance to generate rhythmic structures, or can use the data to feed an adaptive clock and enable a dancer to control musical tempo in real time.

2.1. The Musical Processing of a Dance

As noted by Paul Hodgins [9], music and dance share at the temporal level fundamental characteristics of structure and rhythm so intimately, that is difficult to differentiate each discipline's method of realizing such elements. Choreographer Doris Humphrey [10] considers the "motor" (or "beat") rhythm one of the most important types of rhythm in dance. This rhythm is produced by the motor mechanism, and is the one she considers to be the most important for dancers. On the other hand, the relationship between the perception and production of musical rhythm and bodily activity has already been established (e.g., Fraisse [4]).

This made me realize that it is perhaps possible to analyze certain aspects of dance from a purely musical perspective. By doing so, one would be opening a strictly musical channel of communication between dance and music in interactive dance performance. If a computer can engage in the task of doing a musical analysis and processing of dance movement, it can provide musical elements for interaction to the musician operating the system. The musician would thus be communicating with the dancer in strictly musical terms, in the same fashion she would be communicating with another musician.

It is thus possible, or at least conceivable, to look for these musical elements at the temporal level in dance movement. The software that was developed tries to identify musical elements at the rhythmic level in dance — rhythms present in dance movement that bear the qualities of musical rhythms (i.e. movement articulations that last between 200 and 1800 ms). These musical rhythms, or *musical cues*, are extracted from a movement sequence by performing certain types of frequency-domain analysis on a representation of the movement done by a time-varying representation of the frame-differencing signal.

As I noted earlier [11], the frame-differencing analysis of a video stream can well depict the periodicity in movement, under controlled conditions: fixed camera, controlled lighting, and there needs to be a good contrast between the dancer's image and the background.

a)

b)

c)
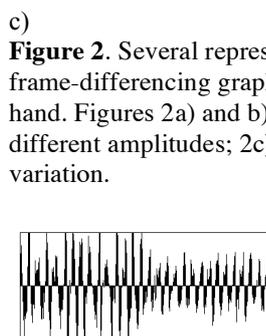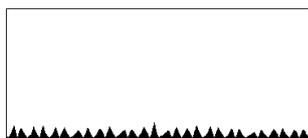
Figure 2. Several representations in the time domain of frame-differencing graphs of movement of a waving hand. Figures 2a) and b), same frequency, with two different amplitudes; 2c), faster frequency with amplitude variation.

Figure 3. DC offset removal of the quantity of motion variation in a video caption of a waving hand.

The library that was created, performs a spectral representation of the time-domain representation of the frame-differencing signal and does certain types of musical analysis on this signal in order to enable a musical channel of communication between dancers and musicians in computer-mediated collaborations. The computer thus acts as a mediator in the process and is responsible for providing the musical elements extracted from movement that can be used for interaction. It acts as a sort of a filter that can extrapolate musical elements from a dance, thereby opening a channel for musical communication/interaction between dancers and electronically-generated music in real time. This situation calls for a more intense participation of the composer/musician in the performance of dance with electronic music. It should also open interesting possibilities for improvisation or for the performance of open-form music/dance structures. The musician operating the computer can always change the parameters for interaction during performance and feed back different musical content to the dancer, or map the musical elements extracted from the dance to different parameters in the electronic score.

3. THE LIBRARY

The library contains six objects: **m.bandit**, **m.clock**, **m.weights**, **m.peak**, **m.sample**, and **m.lp**. These objects can be combined in several ways to perform the temporal domain analysis of dance movement, and can be grouped into objects that do analysis (**m.bandit**, **m.peak**, **m.weights**), processing (**m.clock** and **m.lp**), and extras (**m.sample**). **m.bandit** and **m.clock** are the core of the

library since they are responsible for doing the frequency domain representation of the frame-differencing signal and musical tempo control respectively. Except for **m.lp** that is a subpatch, all objects were coded in C following the published guidelines for writing external objects for Max/MSP.

3.1. Analysis objects

3.1.1. m.bandit

m.bandit is one of the core objects of the library. This object takes as input the time-varying representation of the frame-differencing signal and: (a) estimates and outputs the fundamental frequency of that signal, (b) outputs the amplitude and frequency of the initial four harmonics of that frequency, and (c) outputs lists for graphically displaying the signal spectrum and the spectrum of the four harmonics of the fundamental frequency.

m.bandit also allows the user to specify the object's sensitivity to movement. This is done by sending two messages to the object, 'idle_thresh' and 'idle_int,' which respectively specify the threshold value in the brightness difference below which the object considers that there is no movement and the minimum time interval after being below that value in which the object should consider that there is no movement. The object prints the word 'moving' in the Max window when it considers there is movement (referred to in the text as 'moving state'), and outputs 1 from the rightmost outlet, and prints the word 'idle' (referred to in the text as 'idle state') and outputs zero from the rightmost outlet otherwise. This is independent from the quantity of movement measurements provided by the frame-differencing algorithm. This enables the user to program the object to send control information to other objects only when the measurements of the quantity of movement pass a certain value. One useful and practical application of this feature is, for example, to allow the dancer to move in the backstage without triggering any events while preparing to start a piece.

3.1.2. m.peak

m.peak takes as input the time-varying representation of the frame-differencing signal and outputs a 'bang' message (order of execution) when a significant peak in the signal has occurred. This object is especially useful to trigger events (sounds, or messages to other objects, for example) when heavily accented movement actions occur.

3.1.3. m.weights

m.weights takes as input the fundamental frequency of the frame-differencing signal as calculated by **m.bandit** converted to milliseconds, rounds that value to the hundredth millisecond, and stores it temporarily. This object outputs the hundredth millisecond value that was output more times by **m.bandit** in the past 60 frames. One of the functions of **m.weights** is to provide a "short-term memory" to **m.clock** that can be used to find the

tempo of a movement sequence without a previous estimate. **m.weights** is also a good object to use during performance as a means of permanently monitoring the output of **m.bandit** and suggest tempi values to **m.clock.**

3.2. Processing objects

3.2.1 m.clock

m.clock is the other core object of this library. It is an adaptive clock that enables a dancer to control the musical tempo of a musical sequence. This object treats the fundamental frequency output of **m.bandit** (converted to milliseconds) as a beat candidate and allows to perform the tempo adaptation of a musical sequence if the beat candidate falls within a pre-defined margin for adaptation from a previous tempo estimate.

3.2.2. m.lp

m.lp is a simple low-pass filter that can be used to smooth brightness values computed by the frame-differencing algorithm.
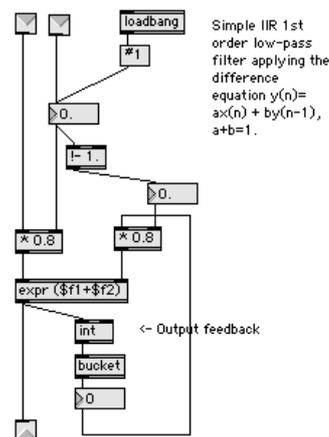
Figure 4. Internal structure of **m.lp**.

4. EXAMPLES AND DEMONSTRATION

All the examples presented for musical rhythm generation and musical tempo control from dance movement assume that the input to the objects is the sum of the absolute brightness change in between frames (such as the value that is sent by object **v.sum** in Figure 5). In the demonstration session I will show how these objects can be connected in order to enable a dancer the musical rhythm generation and musical tempo control in real time in interactive dance performance. Three types of situations utilizing this software in interactive dance performance will be demonstrated: the generation of musical rhythm from the output of **m.bandit**, the control of musical tempo with an initial estimate of the tempo value, and the control of musical tempo without a previous estimate.
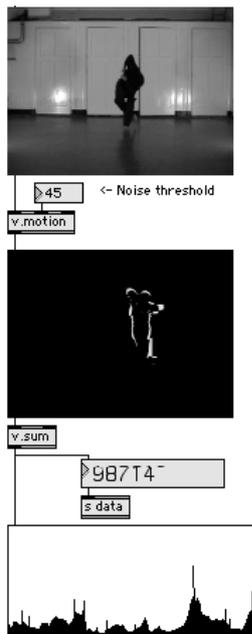
Figure 5. Basic patch for movement capture that sends data to be analyzed by the m-objects.

5. CONCLUSION

This paper presented the m-objects library, a Max/MSP library to be used in interactive dance performance that enables the generation of musical rhythm or the control of musical tempo from dance movement in real time. I introduced the main conceptual framework that underlies the creation of the software, which includes the concept of *musically processing a dance*. This type of processing movement in dance promotes a channel of musical communication between musicians and dancers in computer-mediated performances. I then presented the objects comprising the library.

6. ACKNOWLEDGMENTS

7. REFERENCES

[1] http://www.cycling74.com/products/dlmaxmsp.html

[2] http://www.troikatronix.com/isadora.html

[3] http://www.eyesweb.org/

[4] Fraisse, Paul. Rhythm and tempo. In D. Deutsch (Ed.), *The psychology of music* (pp. 149-180). Orlando, FL; London: Academic Press, 1982

[5] Fraisse, Paul. *Les structures rhythmiques*. Louvain, France: Publications Universitaires de Louvain, 1956

[6] Parncutt, R. The perception of pulse in musical rhythm. In A. Gabrielsson (Ed.), *Action and Perception in Rhtyhm and Music* (pp. 127-138). Stockholm: Royal Swedish Academy of Music, 1987

[7] http://www.cycling74.com/products/dlcyclops.html

[8] http://homepage.mac.com/davidrokeby/softVNS.html

[9] Hodgins, Paul. *Relationships between score and choreography in twentieth-century dance: Music, movement and metaphor*. London: Mellen, 1992

[10] Humphrey, Doris. *The art of making dances*. New York: Grove Press Inc., 1959

[11] Guedes, Carlos. Controlling musical tempo from dance movement: A possible approach. *Proceedings of the International Computer Music Conference*, 453-457. Singapore, 2003