Controlling Musical Tempo from Dance Movement in Real-Time: A Possible Approach

Carlos Guedes

New York University
email: carlos.guedes@nyu.edu

Abstract

In this paper, I present a possible approach for the control of musical tempo in real-time from dance movement. This is done through the processing of video analysis data from a USB web cam that is used to capture the movement sequences. The system presented here, consists of a library of Max externals that is currently under development. This set of Max externals processes video analysis data from other libraries and objects that already do this type of analyses in real-time in this programming environment such as Cyclops (Singer 2001) and softVNS2 (Rokeby 2002). The aim for the creation of such system is to enable dancers the control of musical tempo in real-time in interactive dance environments. In this session I will also do a short a demonstration of the performance of the objects created so far.

1 Introduction

Musical rhythm bears a strong relationship to the physical characteristics of the human body (Parncutt 1987; see also Fraisse 1974; Fraisse 1982). Accompanying music that has a strong sense of pulse with simple body movements such as rocking is a natural human manifestation (Fraisse 1974).

Dance is commonly set to music. The degree of synchronization that can exist between bodily movement in dance and music suggests that there may be some common features between musical rhythm and rhythm in dance. In one of the few studies that address this aspect, Hodgins (1992) notes that analyzing the temporal interaction between dance and music is a difficult task, as the nature of their realizations in the temporal domain is so similar. He also notes the qualitative differences between the rhythmic realizations of music and dance: the gestural tempi of music performance involve in general actions of smaller body parts than those of dance. The degree of temporal accuracy in musical performance may reside in this fact.

However, this does not prevent us from considering that there may be a point of intersection between musical rhythms and the rhythms in dance, since synchronizing bodily movement with musical rhythm is such a natural task for humans.

The considerations summarized above provide the background for the creation of an interactive system that enables dancers to control musical tempo in real-time The main motivations that underlie the creation of such system reside in the fact that, in performances of dance to pre-recorded music, it is sometimes hard for a dancer to maintain proper synchronization with the music. Pre-recorded music imposes a straight jacket on a dancer. Having an expert system that allows a dancer to slightly control the tempo of the music being played could therefore be an interesting aspect to implement in interactive dance performance.

The system presented here consists of a set of Max externals that can do that with a certain degree of success, by processing movement analysis data gathered from a simple web cam. These externals process movement analysis data performed by libraries for image analysis that recently became available to the Max programming environment such as Cyclops (Singer 2001) or softVNS2 (Rokeby 2002). One of the attractive features of this system is that it can produce some interesting results by using a non-invasive medium such as a web cam, and by applying digital signal processing techniques to movement analysis data.

2 The system

The system can be schematically described as follows:

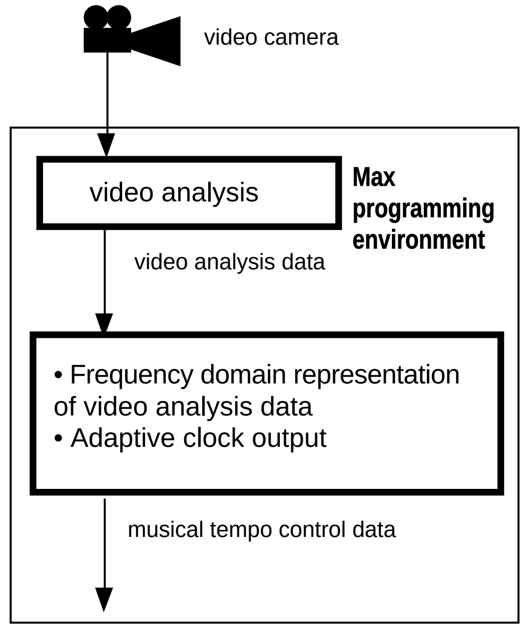

video camera
video analysis
Max programming environment
video analysis data
• Frequency domain representation of video analysis data
• Adaptive clock output
musical tempo control data

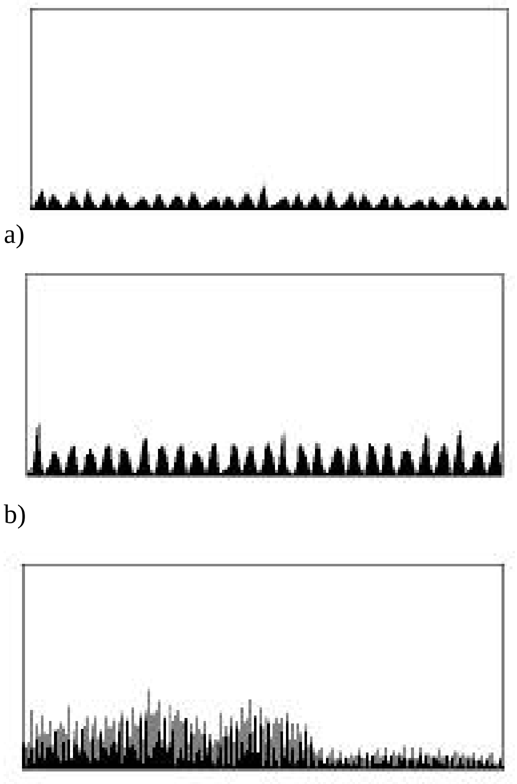
Figure 1. Schematic representation of the system

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 color between consecutive frames are computed (a technique commonly known frame differencing). Subsequently, that time-varying data is given a representation in the frequency domain, and the most prominent frequency is computed. Finally, that frequency value feeds an adaptive clock that can control the tempo of a musical sequence.

2.1 Characteristics of the video analysis signal.

If we are in a relatively controlled lighting environment, we can detect the variation of the *quantity of motion* over time of a moving body by applying frame differencing analysis to the digitized image of that environment. This quantity represents the number of pixels that changed color between consecutive frames. Since the background does not change, all the changes detected through frame differencing will correspond proportionally to the amount of movement performed by the moving body. The more the body moves, the greater will be the number of pixels that changed color between consecutive frames. If we analyze the variation of pixel difference in periodic movement actions over time, we can detect periodicities in the video analysis signal that are in direct correspondence to the actions performed. This works very well for simple periodic actions such as jumping or waving a hand, for example.

Moreover, wider actions correspond to increasing the amplitude of the frame differencing signal and faster actions correspond to increasing the frequency.



a)

b)

c)

Figure 2. Several pixel differences graphs of movement of waving hand. 2 a) and 2 b): Same frequency, with two different amplitudes. 2 c): faster frequency with amplitude variation.

If we remove the DC offset from the signal, the similarities to periodic acoustic signals are simply striking. This means that if we apply some pitch detection algorithm to this signal that works for frequencies in the non-audible range, we can detect the fundamental frequency (tempo) of a periodic movement action.

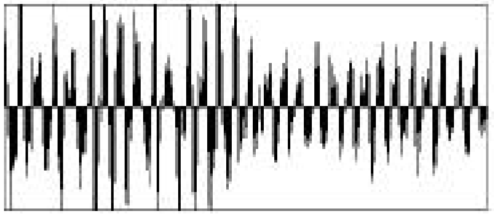


Figure 3. DC offset removal of pixel difference variation of video caption of a waving hand.

2.2 The m-objects

The m-objects are a library of Max externals I am creating for detecting periodicities, including tempo, in dance movement. These objects take as input video analysis data for processing.

This library has at its core two Max externals: **m.bandit**[1] and **m.clock**. **m.bandit** is a bank of 150 second order recursive band pass filters with center frequencies ranging from 0.5 to 15Hz. This bank of band pass filters outputs a frequency domain representation of the video analysis signal, the most prominent frequency detected in the signal, and zero crosses of the phase of the most prominent frequency. **m.clock** is an adaptive clock that can adapt to tempo changes according to some rules. Other objects that help the processing of periodicities are being developed. In this session I will focus mostly on **m.bandit** and **m.clock**.

2.3 Tracking the musical tempo

Musical tempo tracking utilizing adaptive oscillator models or adaptive filters is not new (see for example Large and Kolen 1994, Toiviainen 1998, Cemgil et al 2000). As noted by Rowe (2001), pulse is essentially a form of oscillation, and beat tracking is equivalent to finding the period and phase of a very low frequency. The adaptive model for musical tempo detection and control in dance presented here is inspired on these approaches. Instead of using adaptive oscillators or adaptive filters, tempo detection is done by correlating the frequency-domain representation of the time varying signal with a 1Hz pulse train.

Obtaining the frequency domain representation of the pixel difference variation. One of the functions of **m.bandit** is to give a frequency domain representation of the signal for analysis. In order to obtain a frequency domain representation of the variation over time of the pixel difference values, a bank of 150 second order recursive band pass filters is used. The center frequencies of such filters span from 0.5 to 15 Hz and their bandwidth is proportional to the center frequency (about 10% of the center frequency). Once the pixel difference signal passes through the filter bank, we get a real-time representation of that signal in the frequency domain.

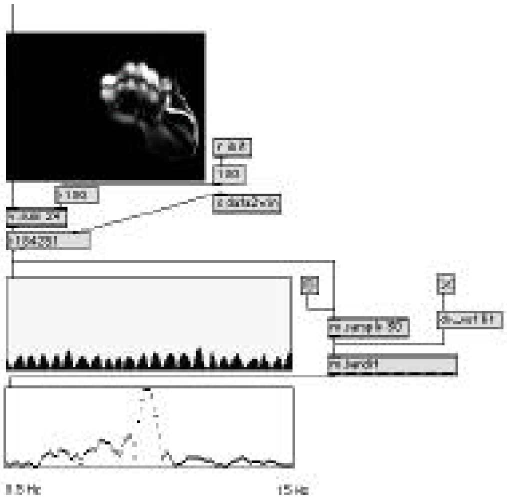


Figure 4. Frequency domain representation of pixel difference variation in movement of a waving hand.

Obtaining the fundamental frequency. In order to obtain the most prominent frequency in the signal, equivalent to the beat, each sample of the frequency domain representation is correlated with a the frequency domain representation of a 1Hz pulse train. The most prominent frequency is obtained by finding the center frequency of the band pass filter that has the highest correlation with the pulse train.

The adaptive clock. m.clock is an adaptive clock that outputs and adapts to the tempo according to certain rules. The most prominent frequency is output by **m.bandit** every frame, i.e. 25 or 30 times per second depending on the amount of frames per second being grabbed.

The adaptive clock is modeled according to the formula:

$$T_n= a*T_{n-1} +b*(T_{n-1}+\Delta t) \qquad (1)$$

T_n is the clock value in milliseconds at frame n, T_{n-1} is the clock value at the previous frame, and Δt is the difference between the measurement that was output by the band pass filter bank (converted to milliseconds) and T_n. A and b are coefficients between 0 and 1, and b=1-a.

This clock only works for values that can be considered musical beats — 300 to 1500 milliseconds (Rowe 2001), or 3.33 to 0.66 Hz. Each time a new value is received, the clock object checks if the value is within the musical beat boundaries. If the value is not within boundaries, that value is ignored and no

[1] Max objects appear in bold typeface in this text

calculations are performed. If the value is the first one to be within boundaries, T_n gets initialized to that value. For subsequent legal beat values, the clock object checks if the variation between the received value and the current clock time is within the allowable margin for variation. If it is, the new clock value is computed according to equation 1[2]. The coefficients a and b can be used to set the degree of "strictness" of the clock. If the user wants the clock to be extremely strict, coefficient a can be set to a value close to 1. This will make the clock very little sensitive to tempo changes induced by the dancer. If, on the other hand, coefficient a is set to a value close to 0 (b=1-a), the clock will adapt faster to tempo changes. The clock object thus has two parameters that can be set by the user. The first, is the value of the allowable margin for variation from the initial beat value. The second value is the degree of strictness of the clock.

This is intended to enable the user of the system to choose the behavior of the clock according to the situation in which is utilized. If the margin and strictness parameters are set to a low value and a high value respectively, the clock will resist a lot to tempo changes behaving almost like a metronome. If the opposite happens, the clock will follow the dancer's tempo behaving almost chaotically. An intermediate situation between these two extremes usually offers the best results.

3 Demonstration

For the demonstration in this session, I built a Max patch that utilizes some softVNS2 objects to do the frame differencing of the video stream. Two situations are presented. The first, demonstrates the performance of the system utilizing live input from a web cam. The second situation shows the performance of the system utilizing short video clips of a dancer dancing samba. The video analysis part of the patch utilizes the objects **v.movie**, **v.dig**, **v.motion**, and **v.sum**. **v.movie** reads and plays a QuickTime movie and outputs the raw video stream data to the **v.motion** object. The object **v.dig** digitizes the input coming of the web cam. **v.motion** performs frame differencing on the video stream, and the object **v.sum** calculates the sum of pixels that change color between consecutive frames. The output from **v.sum** passes through **m.sample** that samples the data at a rate defined by the user. This is intended to optimize the performance of **m.bandit** whose calculations depend on the sampling rate. Finally, the output of **m.bandit** is sent to **m.clock** which in turn sends its output to Max's metro object.

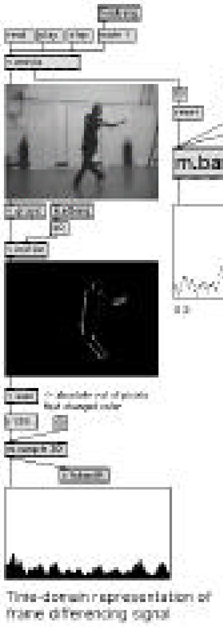

m.bandit
Most prominent frequency ("fundamental")
1.33 Hz
m.clock
753
Tempo (ms)
Time-domain representation of frame differencing signal

Figure 5. Demo patch processing a video clip of choreographer/dancer Susanne Ohmann dancing samba

4 Conclusion

Detecting tempo in dance through the analysis of the variation of pixel differences between consecutive frames of a video stream seems to be a promising way towards enabling dancers to control musical tempo in real-time in interactive dance systems. The fact this system produces some good results in both simple movement sequences such as waving a hand or jumping, as well as in dances containing movement sequences that are well articulated in time, provides a good motivation towards continuing investigating the processing of video analysis signal for tempo detection in movement sequences. The fact that this can be achieved through the use of a simple, non-invasive medium such as a web cam, that is easily set up, can make this system a useful tool for interactive dance performance.

5 Acknowledgments

All of this research was possible thanks to the kind support from the Foundation for Science

[2] I thank Ali Taylan Cemgil for suggesting this approach for the clock behavior.

and Technology and the Luso-American Foundation for the Development in Portugal for my PhD studies at NYU.

I also want to thank Professor Peter Pabon from the Institute of Sonology in Hague for his keen advice on DSP techniques and guidance; Ali Taylan Cemgil for his critical input and valuable suggestions on certain approaches to take; Kirk Woolford for lending me his studio in order to perform the tests; and, last but not least, I want to thank to choreographer/dancer Susanne Ohmann for providing beautiful movement sequences for analysis.

References

Fraisse, P. 1974. *Psychologie du rythme*. Paris: PUF.

Fraisse, P. 1982. "Rhythm and Tempo." In *Psychology of Music*. Ed. Diana Deutsch. London: Academic Press.

Hodgins, Paul. 1992. *Relationships Between Score and Choreography in Twentieth Century Dance : Music, Movement and Metaphor*. London: Mellen.

Large E. and J. Kolen. 1994. "Resonance and the perception of musical meter." *Connection Science* 6:177-208.

Cemgil, A. T., Kappen, B., Desain, P. and H. Honing. 2000. "On Tempo Tracking: Tempogram Representation and Kalman Filtering." *Proceedings of the International Computer Music Conference.* International Computer Music Association.

Parncutt, R. 1987. "The Perception of Pulse in Musical Rhythm." *Action and Perception in Rhythm and Music*. Publications issued by the Royal Swedish Academy of Music No. 55:127-138.

Rokeby, D. 2002. softVNS2. Software: video analysis objects for the Max programming environment.

Rowe, R. 2001. *Machine Musicianship.* Cambridge MA: MIT Press.

Singer, E. 2001. Cyclops. Software: Max external.

Toiviainen, P. "An Interactive MIDI Accompanist." *Computer Music Journal* 22(4):63-75.