# A Wii-based gestural interface for computer-based conducting systems

**Lijuan Peng**
Computer Science Department
University of Regina
SK, Canada, S4S 0A2
peng200l@uregina.ca

**Dr. David Gerhard**
Computer Science Department
University of Regina
SK, Canada, S4S 0A2
david.gerhard@uregina.ca

## Abstract

As the increase of the sales of Wii game consoles, the Wii Remote is able to used as a common input device for a computer system. Some software is developed for acquiring the data from the Wii Remote for further processing. This paper presents a Wii-based gestural interface for computer-based conducting systems. It employs the infrared camera in a Wii Remote to capture the conducting gestures of a conductor who is holding an infrared baton, and then represents the gestures using visual and aural representations after data analysis and gesture classification/following. The interface is intended to be used for pedagogy purposes. So it accepts standard conducting gestures, uses an infrared baton similar to a real baton and displays the trajectory of a gesture on the screen which can be compared to the corresponding diagram shown in a textbook. In addition, aural feedback is also supported in this interface. It plays a MIDI note to represent a certain beat in a conducting gesture.

**Keywords:** Aural representation, Computer-based conducting systems, Gestural interface, Infrared baton/camera, Pedagogy, the Wii remote, Visual representation.

## 1. Introduction

Computer-based conducting systems [**?**][**?**][**?**] allow a user to conduct a piece of music using a digital system. Most of them are manipulated using a gestural interface. The gestures can be a standard conducting gesture or a simplified version, such as up and down.

The gestures in a gestural interface have to be captured and transfered into a series of sample data for further processing. Various sensors have been employed in previous computer-based conducting systems, for example a camera. The Wii Remote, as a new type of common input device, can also be used in such a system [**?**] because of its motion sensing ability and the infrared camera set on the top. This paper presents a Wii-based gestural interface for computer-based conducting systems. It is intended to be used for pedagogy purposes.



## 2. Background and Related Research

This section describes the related background, including the Human Computer Interface (HCI), gestural interfaces in musical systems, conducting, and interfaces in computer-based conducting systems. After that, a brief introduction of the Wii Remote is presented.

### 2.1. Human Computer Interface

Here, Human Computer Interface (HCI) is a broad term including not only the Graphical User Interface (GUI), but also any interaction between a user and a computer system. Various interfaces have been designed and developed over the past few years, such as, Graphical User Interface (GUI), Conversational Interface Agents, Gestural Interface and so on.

Figure **??** demonstrates the relationship between an interface and a computer system. A user manipulates a computer system via certain input devices. Then, the input data is processed in functional modules and results are generated and presented back to the user.

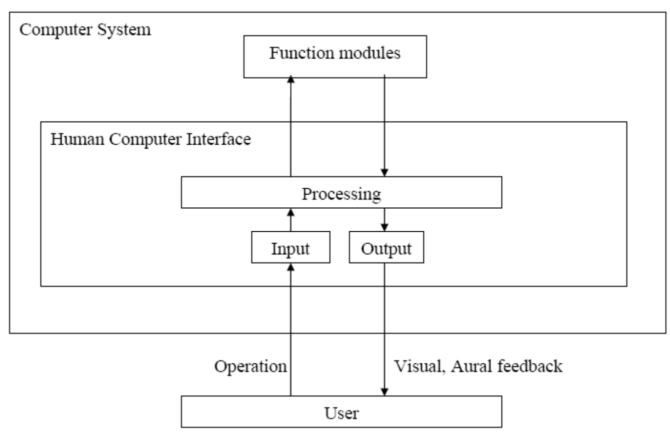



Figure 1. The relationship between the interface and the computer system

### 2.2. Gestural interface in musical systems

Gestural interfaces have been used in musical performance, composition, and conducting systems. Almost all conducting systems employ a gestural interface. Applications related to a certain instrument may be used in a way that is similar to playing the corresponding acoustic instrument such as *Le SuperPolm* [?]. Furthermore, gestural interfaces can inspire creativity in musicians or provide opportunities for the public to join in music activities (e.g. improvisation). *BodyMusic* [?] is such a system, allowing the composition of music using hand gestures.

In most cases, gestures cannot be entered using a keyboard and mouse, the most common input devices for a computer. Generally, gestures are captured by sensors such as a video camera or acceleration sensor. After processing, gestures are mapped to some musical parameters and are used to control the response to users. For example, in *BodyMusic* [?], hand gestures, which are acquired using a pair of data gloves and a tracker, control the melody of a music piece composed by a user. A sound synthesizer or sequencer is involved in a lot of musical systems for the sake of an inherent characteristic to generate sound effects or musical pieces. Therefore, it is not surprising that most musical systems give out aural feedback to users.

### 2.3. Conducting

Generally, conducting is performed in the conducting window, a chest-high and virtual rectangular area. The movements, which can be up, down, left, and right, follow standard conducting patterns and may be continuous or have a stop between two motions. Legato and staccato are two of the primary types of beat-patterns. Legato consists of continuous and curved motions. Staccato has a stop in a moment at each count. It contains relatively straight motions. [?]

### 2.4. Interfaces in computer-based conducting systems

The first computer-based conducting system, the *Microcomputer-Based Conducting System* [?], was developed in 1980 and did not implement a gestural interface. It accepted an input using a graphics tablet, switches, or slides.

After that, researchers began to utilize gestures to manipulate systems. Acceleration sensors [?], cameras [?], infrared sensors [?], and many other sensors, such as the Wii Remote [?], have been employed to track the gestures of a conductor. And then, aural and/or visual response is presented to a user after data analysis and gesture classification or following. In a conducting system, aural response, which may be either MIDI or audio, is an essential response sent to users. Video [?] or animated virtual orchestra [?] are supplementary feedback to sound. Only one system, the *Virtual Conducting Practice Environment* [?], provides visual representation of conducting gestures for conducting students such as beat windows.

### 2.5. The Wii Remote

The Wii Remote is a controller for Nintendo's Wii game system. Figure **??** shows a picture of the Wii Remote from Nintendo's website.


**Figure 2. Wii Remote [?]**

Generally, it is used as a game controller. It has a motion sensor that can acquire 3-dimensional acceleration data. The acceleration sample data is able to be shown in a graph using an OS X software named DarwiinremoteOSC [?].

In addition, the Wii Remote has an infrared camera set on its top. The infrared camera tracks up to 4 infrared sources. Thus, the Wii Remote can be one of input devices of a computer system. Johnny Chung Lee [?] has applied the infrared camera in a Wii Remote in several projects.

## 3. Implementation

Figure **??** displays the concepts contained in this system. The gestures captured by a Wii Remote has to be processed. The generated results should be transfered into a proper representation for users. Function modules in the corresponding computer system complete simple or complex tasks, such as data analysis and gesture classification.

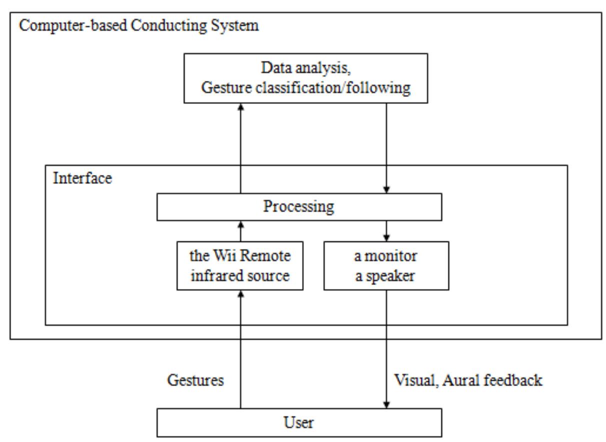



**Figure 3. The concepts contained in this system**

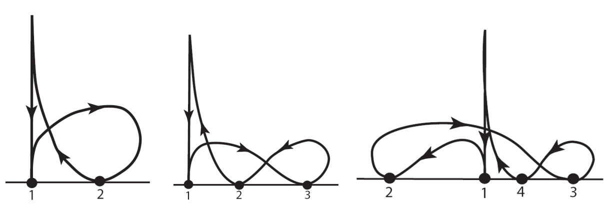
The whole system is run on a iMac personal computer. Figure **??** shows the setup of the system. A user is standing in front of a computer and the Wii Remote is set on the left side of the computer. An infrared baton is held in the user's right hand. Actually, the position of the Wii Remote can be adjusted by users. Most gestures should be in the field of view of the Wii Remote. Meantime, users are comfortable during conducting.

**Figure 4. The setup**

This system is developed using Max/MSP [**?**], Jitter [**?**] and Java. Max/MSP is a graphical programming environment for real-time multimedia applications. Jitter is an extension to Max/MSP specifically designed for graphic manipulations and is very valuable for developing realtime video applications and processing matrix data. The Java programming language is utilized to develop functions that Max/MSP does not support or are difficult to implement such as data sorting.

### 3.1. Gestures

A gestural interface makes use of diverse gestures to control computer systems. Hand gestures, body movements, and even facial actions can be used. Conducting gestures contains right arm movement, left arm movement, eye contact, breathing, and other gestures. In this interface, expressive legato gestures are supported. Figure **??** [**?**] shows three beat patterns: 2 to 4 beats per measure.




(a) 2-beat pattern (b) 3-beat pattern (c) 4-beat pattern

**Figure 5. Three beat patterns**

### 3.2. Wii Remote

In this gestural interface, the Wii Remote is used as an infrared camera to track the movement of the right hand. DarwiinremoteOSC receives sample data from the Wii Remote via a bluetooth connection. The sample data contains the coordinates of an infrared source. After that, DarwiinremoteOSC transfers infrared data into Open Sound Control (OSC) [**?**] messages. OSC is a protocol for communication among multimedia applications similar to MIDI. These OSC messages are sent out by DarwiinremoteOSC to gesture analysis for further processing via the User Datagram Protocol (UDP), one of the core protocols of the Internet. Because UDP does not check the delivery of each packet, it is fast and suitable for time-sensitive applications.

Using the Wii Remote, users do not need to pay for a specific sensor bought or developed only for this gestural interface. Max/MSP supports both the OSC protocol and UDP, so it is easy to acquire the sample data with the help of the DarwiinremoteOSC. More importantly, the infrared camera only captures the movement of infrared sources and has no specific requirements for the background. Therefore, it is suitable to be used in a natural environment.

One problem for the Wii Remote is its field of view. The horizontal and vertical field of view are approximately 31 degrees and 41 degrees, respectively. A motion beyond this range cannot be tracked and will be lost. Thus, the position of the Wii Remote has to be adjusted in accordance with the position of users before conducting.

### 3.3. Infrared baton

During conducting, an infrared baton is held in the right hand of a conductor as an infrared light source. It consists of a conducting baton, a button, a one-cell battery holder, a 1.5v battery and an infrared LED as shown in Figure **??**.

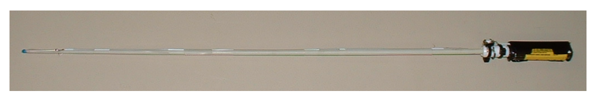


**Figure 6. An infrared baton**

The infrared LED used in this gestural interface is chosen because of its large viewing angle of 110 degrees. If the viewing angle and radiant intensity are too small, the tracking will be easily lost while conducting. While using this baton, users have to hold it in their right hand and press the button with their thumb. The infrared LED on the tip of the baton will emit infrared radiation, which is captured by the infrared sensor in the Wii Remote. During conducting, the trajectory of the infrared LED reveals the beat pattern conducted by users. This baton can also be employed as a mouse. The position of the baton helps users choose from one of five options. This infrared baton is a compact and light weight controller. It looks very much like a real baton. Actually, it is made of a real baton with the handle removed. A one celled battery is equipped at the bottom of the baton where the handle was. Thus, it is slightly longer than a real baton. The weight of this baton derives mainly from the 1.5v dry battery and is therefore heavier than a real baton. During conducting, the difference from a real baton is that the button on the infrared baton has to be pressed by the thumb of a user. This button controls the connection between the infrared LED and the battery. However, while holding a real baton, the thumb of a conductor has to be put on some position of the baton with a little strength. Thus, it may not affect the conducting of users except that a little more strength required.

The technical problem for this infrared baton is its viewing angle. Although 110 degrees is a very large viewing angle, it still may lose some tracking data if too large a motion is used. However, this is not a big problem because users rarely conduct with such a large gesture, and further, such a large conducting gesture would be incorrect. A loss of tracking data can therefore be considered an incorrect gesture.

### 3.4. Main window

The main window is shown on a monitor. It displays the menu, the conducting window, the results of gesture classification/following, and other related information. The menu contains five options to support two types of gesture recognition and the gesture following. Any option can be chosen using the infrared baton, while the focus is in the menu area. The conducting window displays the trajectory of the infrared baton and beat information, while it contains the focus.

Two versions have been developed for the main window. One invovles the infrared baton and a mouse. The other only uses the infrared baton because it can also work as a mouse. Figure **??** and Figure **??** are two versions of the main window.

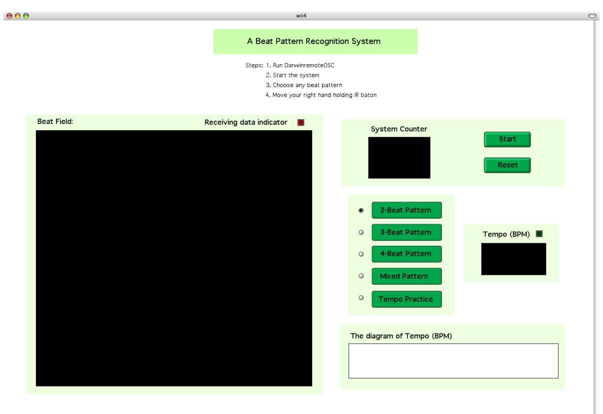



**Figure 7. The first version of the main window**

The most important difference between them is the switch between menu selection and conducting. While using the first version, a user has to manipulate two input devices, an infrared baton for conducting and a mouse for selecting options. Thus, an extra time is required for the switch, such as the time to lay down the mouse and grasp the infrared baton. Whereas, a uniform controller, the infrared baton, is employed in the second version to switch between two modes, the option selection mode and the conducting mode.

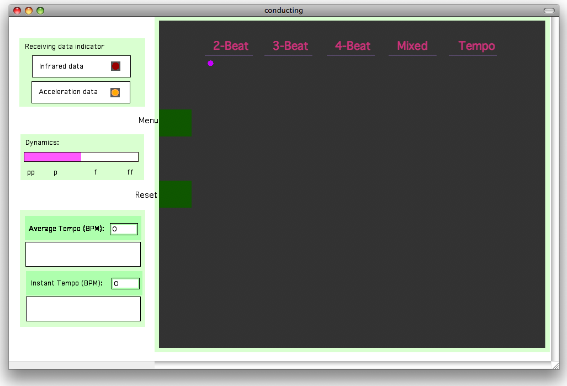



**Figure 8. The second version of the main window**

The focus is indicated by the color of a short bar, "Menu". The brighter color represents the option selection mode. The conducting mode corresponds to a darker color. Thus, the "Menu" bar mimics a switch button. It is designed for going back the menu area (the option selection mode). There is no specific switch bar in the menu. If a user holding the infrared baton keeps on an option, for example 2-Beat, for a period of time, the focus will be moved back to the conducting window (the conducting mode). A uniform controller eliminates the extra time spent on switching between a baton and a mouse and provides a smooth operation of the system. Therefore, the second version is chosen for this gestural interface.

### 3.5. Visual representation

Visual representation of a conducting gesture is displayed on the main window. It is a straightforward explanation of a conducting gesture.

In this gestural interface, the movement of the infrared baton is drawn on the conducting window. It consists of small dots and a curve. The small dots are coordinates of the infrared baton. The curve connects all small dots and represents the trajectory of the infrared baton (the right hand). Figure **??** shows a trajectory of the movement of the right hand. It seems that the right hand is drawing a circle.

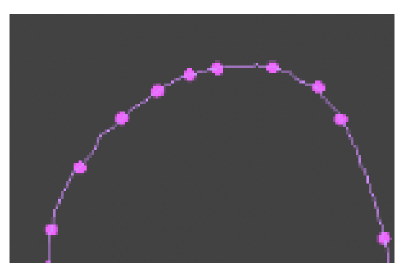


**Figure 9. A trajectory of the movement of the right hand**

In this system, several modules are implemented to support gesture classification and following. Thus, more visual items can be displayed on the conducting window besides the visual representation of input data (the trajectory). They includes beats, numbers of beats, and the result of gesture classification and following as shown in Figure **??**. Compared with Figure **??**, a user may be able to figure out how to improve his/her gesture.

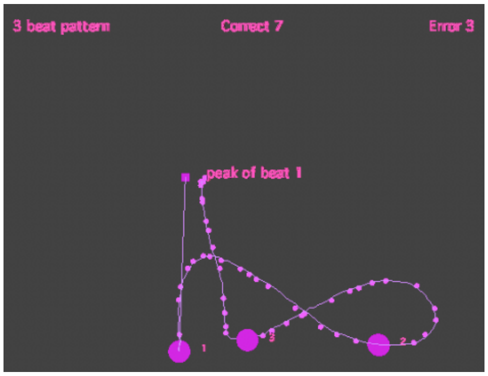



**Figure 10. A beat pattern performed by the right hand**

The result of gesture following, tempo, is represented using both a numerical value and a diagram. Figure **??** displays an example. It reveals the speed of conducting.

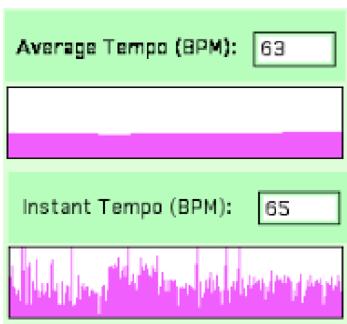



**Figure 11. Visual representation of the value of tempo**

### 3.6. Aural representation

This system also supports aural representation for a conducting gesture. An individual MIDI note is associated to a certain beat. Once a beat is found, the corrreponding MIDI note is played, such as C4 for the downbeat (the first beat).

In addition, G7 is used to represent ten incorrect conducting gestures because its pitch is much higher than the MIDI notes used for beats. Therefore, it is easy for a user to realize too much errors occur. They can adjust their gestures and re-start again.

### 3.7. Data analysis, gesture recognition, and gesture following

In this system, a few modules including data analysis, gesture recognition, and gesture following are implemented. They are on the basis of input data. Geture recognition groups conducting gestures into different categories. Gesture following passes the processed input data to output.

The data from the Wii Remote are coordinates of the infrared LED on the tip of the infrared baton. Data analysis picks up the vertical maxima and minima from these coordinates and passes them to gesture recognition and following. The reason to look for vertical minima is that each beat occurs at the vertical mimimum according to inherent characteristics of conducting gestures [**?**] [**?**].

This system has two types of gesture classification. A general beat recognition is implemented that counts beats but does not identify whether or not the beat pattern is correct. Beat patterns between 2 and 12 beats are supported. Another mode is a specific one that only supports three beat patterns (from 2 to 4 beats per measure). It is intended to figure out the correctness of a gesture according to a desired pattern. Thus, the result of gesture classification is either a certain beat pattern or an an error gesture. Figure **??** shows a conducting gesture that is identified as a 3-beat pattern.

This system also supports tempo tracking. It is intended to reveal the spead of the conducting. Both average and instant tempo are implemented. The value of the tempo is calculated using Equation **??** and Equation **??**.

$$Tempo_a = \begin{cases} \frac{N_b}{T_b}, & if\ N_b < 10, \\ \frac{10}{T_b}, & if\ N_b \geq 10. \end{cases} \quad (1)$$

$$Tempo_i = \frac{1}{T_{one}} \quad (2)$$

Among them, $Tempo_a$ and $Tempo_i$ are the average value and the instant value of the tempo, respectively. $N_b$ denotes the number of beats. $T_b$ and $T_{one}$ represent the duration to produce $N_b$/10 beats and one beat, respectively. As shown in Equation **??**, the average value is a moving average that is estimated based on the past 10 beats if more than 10 beats have been conducted. Therefore, it can follow the changes, especially the significant changes, in tempo quickly and also show the speed trend of conducting clearly.

## 4. Conclusion

This paper presents a gestural interface that uses a Wii Remote and an infrared baton to track conducting gestures of the right hand. Visual and aural representation are supported. The Wii Remote is a new input device for a computer system and able to support motion tracking with its motion sensor and infrared camera. Visual representation is a straightforward explanation of gestures and a supplementary of aural representation for a musical system. In addition, data analysis, gesture classification and gesture following are implemented to support more functions instead of just tracking and representing gestures.

This gestural interface can be used for pedagogical purposes, such as learning and practice of beat patterns. In future, user testing will be conducted. More functions will be designed and developed, such as support of Staccato beat patterns.

## References


[1] W. Buxton, W. Reeves, G. Fedorknow, K. C. Smith and R. Baecker, "A microcomputer-based conducting system", *Computer Music Journal*, Vol. 4, No. 1, pages 8-21, 1980.

[2] "Max/MSP/Jitter" [Web site] 2009, [2009 Jan 11], Available: `http://www.cycling74.com/products/max5`.

[3] "darwiinosc" [Web site] 2008, [2009 Jan 11], Available: `http://code.google.com/p/darwiinosc/`.

[4] Guy E. Garnett, Fernando Malvar-Ruiz and Fred Stoltzfus, "Virtual conducting practice environment", In *Proceedings of the International Computer Music Conference*, pages 371-374. ICMA, 1999.

[5] Suguru Goto and Takahiko Suzuki, "The case study of application of advanced gesture interface and mapping interface", In *Proceedings of the 2004 Conference on New Interfaces for Musical Expression (NIME04)*, 2004, pages 207-208.

[6] Horace H S Ip, Belton Kwong and Ken C K Law, "Bodymusic: a novel framework design for body-driven music composition", In *ACE'05: Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, pages 342-345, New York, NY, USA, 2005. ACM Press.

[7] "Wii Projects" [Web site] 2008, [2009 Jan 11], Available: `http://www.cs.cmu.edu/~johnny/projects/wii/`.

[8] Joseph A. Labuta, "Basic conducting techniques, fourth edition", Prentice Hall, 2000.

[9] Eric Lee, Henning Kiel, Saskia Dedenbach, Ingo Grull, Thorsten Karrer, Marius Wolf and Jan Borchers, "iSymphony: an adaptive interactive orchestral conducting system for digital audio and video streams", In *CHI'06: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 259-262, 2006.

[10] Declan Murphy, Tue Haste Andersen and Kristoffer Jensen, "Conducting audio files via computer vision", In *Proceedings of the 2003 International Gesture Worshop*, 2003.

[11] Teresa Anne Marrin, "Toward an understanding of musical gesture: mapping expressive intention with the digital baton", Master's thesis, Massachusetts Institute of Technology, 1996.

[12] Brock McElheran, "Conducting technique for beginners and professionals revised edition", Oxford University Press, 1989.

[13] "Open Sound Control" [Web site] 2006, [2009 Jan 11], Available: `http://archive.cnmat.berkeley.edu/OpenSoundControl/`.

[14] Satoshi Usa and Yasunori Mochida, "A multi-modal conducting simulator", In *Proceedings of the International Computer Music Conference*, pages 25-32. ICMA, 1998.

[15] "Wii Music Orchestra" [Web site] 2006, [2009 Jan 11], Available: `http://www.gamespot.com/wii/puzzle/wiimusicorchestra/index.html`.

[16] "Wii Controllers" [Web site] 2008, [2009 Jan 11], Available: `http://www.nintendo.com/wii/what/controllers\sharpremote`.