Wiiolin: a violin-emulating quintephone

Jace Miller
Texas A&M University

ABSTRACT

The console gaming industry is experiencing a revolution in terms of user control, due in large part to Nintendo's introduction of the Wii remote. The online open source development community has embraced the Wii remote, integrating the inexpensive technology into numerous applications. Some of the more interesting applications demonstrate how the remote hardware can be leveraged for nonstandard uses. In this paper we describe a new way of interacting with the Wii remote and sensor bar to produce music. The Wiiolin is a virtual instrument which can mimic a violin or cello. Sensor bar motion relative to the Wii remote and button presses are analyzed in real-time to generate notes. Our design is novel in that it involves the remote's infrared camera and sensor bar as an integral part of music production. The Wiiolin introduces a more realistic way of instrument interaction than other attempts which rely on button presses and accelerometer data alone.

INTRODUCTION

Nintendo's Wii has rejuvenated the gaming industry, largely in part to it's revolutionary controller. The Wii remote, or Wiimote as some have informally termed it, comes equipped with accelerometer sensors and an infrared (IR) camera. Leveraging this affordable technology opens up the door for new applications. This work describes a new way of interacting with the Wii remote and sensor bar to simulate playing a virtual string instrument.

Figure 1: The orientation of the Wii remote determines wheter the Wiiolin performs as a violin or cello

RELATED WORK

Instrument simulation through peripherals has been successfully implemented in popular video games such as Guitar Hero, Donkey Konga, and Rock Band [2, 7]. Nintendo is currently developing a new game entitled Wii Music which allows the tempo and pitch of a song to be altered using information gathered from button presses and the controller's accelerometer [5]. Johnny Lee's Head Tracking for Desktop Displays project uses the sensor bar in a nonstandard, "reverse" way – holding the Wiimote fixed and primarily moving the sensor bar [4]. The Wiiolin combines the ideas of instrument simulation and using the sensor bar in a nonstandard way to create a new quintephone – an instrument that generates sound informatically. We are not aware of any other work that attempts to recognize sensor bar movement as a primary means of sound generation.

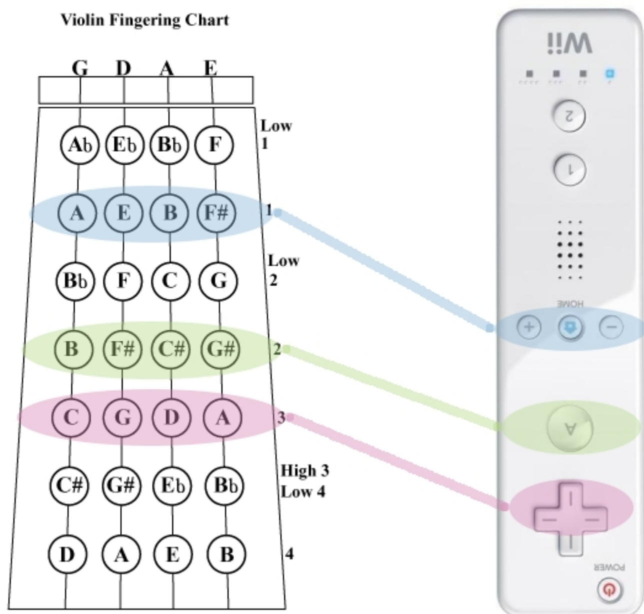

Illustration 1: Mapping violin finger positions to Wii remote buttons

DESIGN

A primary goal for the Wiiolin was the ability to mimic features of music production found in actual string instruments. These features include the idea of bow movement across strings to produce sound, and note pitch being determined by finger position on a string. Another goal was the support of multiple stringed instruments through a similar interface. While MIDI output is often used for computer generated music because it allows easy, instantaneous production of notes with the ability to sustain the note for an arbitrary length of time, we wanted to capture the sound quality reminiscent of the timbre of a string instrument, which can only be achieved through playback of recorded sound samples [6].

The instruments belonging to the violin family, which include the violin, viola, cello, and bass, all have four strings. For each of these instruments, the standard tuning of strings form perfect fifths. Playing the open strings of a violin with no finger positions produces the notes G, D, A, and E. Standard first position, which corresponds to the key of D major, was chosen as the model for mapping finger positions to Wii remote buttons. Incidentally, the relative spacings of the buttons chosen for mapping finger positions are similar to actual standard first position positions. The buttons mapped to first and second finger positions are farther apart than the buttons mapped to second and third finger position.

IMPLEMENTATION

The Wii remote is connected to a computer using a MSI Star Bluetooth USB adapter. Once connected, the Wiiolin program is executed, which continuously loops through performing three main tasks: gathering input from the Wii remote, examining the inputs to determine what note to play, and playing back the correct sound sample. Connecting to the Wii remote programmatically and gathering its raw input values was aided by the use of the Wiiuse C library version 0.12 [8]. The FMOD Ex 4.10 sound library was selected as the means of loading sound files into memory and handling their playback [1]. Determining what note to play consists of analyzing the state of the inputs read from the Wii remote, following the algorithm outlined below:

1. Determine the instrument
2. Determine the string
3. Determine if proper bowing motion is occurring
4. Determine the finger position
5. Select the note to be played

The type of instrument is selected based on data collected from accelerometer sensors. Values of the three accelerometer axes can be analyzed to determine the tilt of the controller with respect to the ground, given the effects of Earth's gravity. When the Wii remote is held horizontally, violin sound samples will be used; but when the the Wii remote is held vertically, cello sound samples will be used. Figure 1 shows the different postures for playing the Wiiolin in its violin and cello forms. The threshold for switching instruments was set as 45° below the horizontal.

Determining the virtual string that should potentially be played is done by examining the positions of points of infrared light produced by the light emitting diodes (LEDs) in the sensor bar and detected by the Wii remote's infrared camera. Only sources of infrared light are visible to the IR camera, which can track up to four points of IR light at a time. The sensor bar is commonly used to provide two sources of IR light, whose positions are plotted in a two dimensional plane perpendicular to the IR camera's line of sight. The slope of a line through these points, which we refer to as the sensor bar slope, is calculated and used as the feature that selects the string. The slope space is partitioned into four sections – one for each string – with three thresholds.

The direction of proper bowing motion is dependent on the string being played. Specifically, the notion of sensor bar slope needs to be understood before bowing motion can accurately be detected. For bowing motion to produce a sound, it should occur along a vector nearly parallel to the sensor bar slope. Otherwise, moving the bow perpendicular to the normal bowing direction would also produce a sound. This erroneous situation can be avoided by finding the projection of the vector of motion onto the sensor bar slope and using its magnitude as the movement speed instead of the magnitude of the vector of movement before being projected. When the projected movement speed of the sensor bar is above a threshold, correct bowing motion is signaled.

Finger positions are determined by which buttons are pressed on the face of the Wiimote. Three buttons are mapped to both first and third finger positions to allow a greater flexibility in hand position. Just as the note produced by a real string instrument is determined by the finger position pressed nearest the bridge, the note produced by the Wiiolin is determined by the button being pressed nearest the infrared camera.

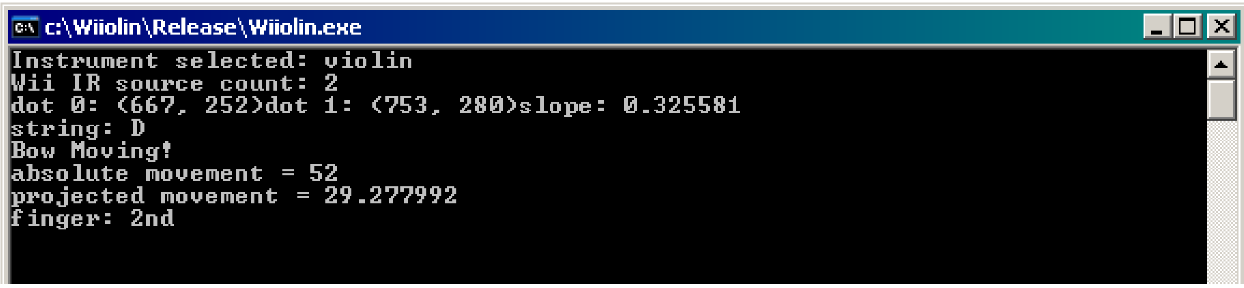

Figure 2: Wiiolin implementation screenshot

The Wii remote's infrared camera has a viewing angle of approximately 45°. This small field of view limits the range that a sensor bar can be moved while keeping both IR sources in the camera's sight. Examining a sensor bar through an IR sensitive camera reveals that each side of the bar actually contained three LEDs. The solution to LED spacing was to cover the center LED on one side of the bar and cover all three LEDs on the other side of the bar with the user's hand. This simple modification to the sensor bar can be seen in Figure 1. Using the closer LEDs reduced the spacing by a factor of eight, allowing a larger range of bowing motion to be detected by the IR camera while the sensor bar remains relatively close to the Wiimote.

Timing the playback of recorded music is difficult for interactive media. The challenges of integrating recorded sound with an interactive environment are just now being addressed in the gaming industry. Super Mario Galaxy, the newest iteration of the Mario franchise, is the first game in the series to use orchestrated music. During the recording of the orchestra, a very strict tempo was maintained by means of a click track so that the music tempo would synchronize with the game tempo [3]. During the development of Wiiolin, we noticed that playing sound samples from the beginning resulted in what could be perceived as a sluggish response. The time delay between the start of the sound sample and the point where it reached its recognizable pitch hampered the interactive feel of the system. To alleviate this problem, sound samples were started at 300 ms after their beginning – about the time they reach their recognizable pitch. This improved the perceived response time, but has the slight drawback of sometimes making transitions between notes sound too abrupt to be natural.

RESULTS

We performed an informal user study on our prototype Wiiolin. Twenty individuals ranging from age 18 to 52 tested and tried to play the Wiiolin. Their abilities to play the Wiiolin were quite varied. All were able to produce notes when using both the violin and the cello configurations, but only those people who had experience playing a violin or cello were able to play recognizable melodies. The users found the cello position to be slightly more easy to play than the violin. We think this is due to the fact that the distance from the infrared camera to the sensor bar is generally greater when the Wiiolin is being played as a cello. All users greatly enjoyed playing with the Wiiolin and wanted to continue experimentation, enjoying the challenge of getting the Wiiolin to play actual songs.

While there is obviously a learning curve to the Wiiolin, the authors are happy with the fact that some of the learning curve maps directly to the learning of a real stringed instrument. Since the Wiiolin uses buttons to map the process of choosing what note to play to a discrete task, the learning curve is much shorter than that of an actual instrument where players must learn to memorize and precisely control continuous finger positions. The discrete mapping of notes improves note selection in a way similar to the frets on a guitar, the advantage being excellent accuracy in note pitch.

The general consensus of all the play testers was that the idea was novel and interesting. Some suggested that the prototype has the potential to be developed into a marketable product. The positive feedback has encouraged continued development of the concept, especially in regards to improving the user experience and extending the functionality of the system.

FUTURE WORK

One of the large differences that we notice between the Wiiolin and a violin is in the physical support of the instrument. A violin is held in place with nothing but the chin and shoulder. The left hand does not hold up the violin, but is used only for counteracting the force of finger presses on the strings. For the Wiiolin, the left hand must support the Wii remote, and keep it steady. Keeping the Wii remote level about its Z axis is made more important since the selected string is determined by the slope of a line in the X-Y plane. A potential solution to this problem would be to build a frame that rests on the shoulder, extends to the left hand, and attaches to the Wii remote to hold it steady. The frame could also be designed with a notch to rest the sensor bar on when bowing, yet does not obstruct the view of the IR camera.

Another feature of a real violin that we have not yet incorporated is the ability to play two strings at once. This technique is called double stopping and produces a chord of two notes. The ability to play double stopped notes could be achieved for the Wiiolin by detecting when the sensor bar slope is very near the currently defined thresholds for distinguishing strings.

Currently, only two instruments are supported, but more could easily be added by incorporating sound samples from different instruments. Although any instrument could be added, the two that most naturally fit the physical mapping of the current setup are the remaining members of the violin family – the viola and the bass. The violin and viola are held more or less horizontally when played, while the cello and bass are held vertically. In addition to Wii remote tilt, the distance from the Wii remote to the sensor bar could be a feature used to automatically determine which instrument should be played. Calculating the distance from the Wiimote's camera to the sensor bar can be accomplished since the distance between the two infrared LEDs is fixed. The higher and lower pairs of instruments could be classified based on Wii remote alignment, while the separation between violin and viola, or cello and bass, can be done by setting a threshold for sensor bar to Wii remote distance. This threshold would most likely have to be a function of where the center of the two IR LEDs lies along the Wii remote's X axis since the distinction of distance between the violin and viola would be fairly small. In general, the sensor bar to Wiimote distance of the viola should be farther than that of the violin. However, the sensor bar to Wiimote distance for the viola when the center of the sensor bar is closest to the Wiimote (when aligned along the Wiimote's Z axis) is less than the sensor bar to Wiimote distance for the violin when the center of the sensor bar is far from the Wiimote (when the IR LEDs are near the peripheral edge of the Wiimote camera's view). Some experimental testing will need to be done before the threshold values needed to separate the violin and viola instruments by distance can be determined.

Other possible improvements include simulating vibrato and using the Wii remote's internal speaker for sound playback. Vibrato is an advanced technique achieved by moving the finger quickly back and forth along the string while maintaining the pressure to produce clear notes. For the Wiiolin, this action could be cued by small back and forth acceleration along the Wiimote's Z axis. Although it is not known for great sound quality, the Wii remote's small internal speaker is supported by the FMOD Ex library and could be used to play the sound samples, removing the requirement of computer speakers.

REFERENCES

1. FMOD Music and Sound Effects System. http://www.fmod.org.
2. Guitar Hero Official Website. http://www.guitarhero.com/.
3. IGN Wii – The Music of Super Mario Galaxy. http://music.ign.com/articles/870/870662p1.html.
4. Lee, J.C. Head Tracking for Desktop VR Displays using the Wii Remote. http://www.cs.cmu.edu/~johnny/projects/wii/.
5. Nintendo Official Website - E3 2008 - Wii Music. http://e3.nintendo.com/wii/wiimusic/index.html.
6. Philharmonia Orchestra Sound Sample Library. http://www.philharmonia.co.uk/thesoundexchange/sound_samples/sample_libraries/.
7. Rock Band Official Website. http://www.rockband.com/.
8. Wiiuse – The Wiimote C Library. http://www.wiiuse.net/.