Teaching Robots to Conduct: Automatic Extraction of Conducting Information from Sheet Music

Andrea Salgian
The College of New Jersey
salgian@tcnj.edu

Lawrence Agina
The College of New Jersey
agina2@tcnj.edu

Teresa M. Nakra
The College of New Jersey
nakra@tcnj.edu

ABSTRACT

While a large number of human computer interaction systems are aimed at allowing the user to conduct a virtual orchestra, very few attempts have been made to solve the reverse problem of building a computer-based conductor that can conduct a real orchestra. The only known instances of robotic conductors had prerecorded performances that require reprogramming for every new musical piece. In this paper we present a family of artificial conducting systems that rely on a novel parsing algorithm to extract conducting information from sheet music encoded in MIDI files. The algorithm was successfully implemented in humanoid and non-humanoid robots and animations and tested in a live concert with student musicians.

1. INTRODUCTION

Recently, human-computer interaction applications have shown an increasing interest towards musical expression, and with the ubiquitousness of computing in all areas, musicians and the public are showing an increased acceptance of computers supplementing classical performances. Among these, conducting is perhaps one of the most targeted, since conductors are the only musicians who freely move their hands to create sound, and whose gestures are not constrained by a rigid instrument.

Several computer-based conducting recognition systems have been developed as interfaces between human conductors and computer-based virtual orchestras. Such environments allow real conductors, or the general public, to conduct by adjusting effects of a prerecorded score, most commonly tempo and volume. The first such system was Mathews' *Radio Baton* [1], which used the movement of a baton emitting radio frequency signals. It was followed by Marrin's *Digital Baton* [2], which, in addition to baton position, used parameters such as pressure on parts of the handle. Usa's *MultiModal Conducting Simulator* [3] used Hidden Markov Models and fuzzy logic to track gestures. Murphy *et al.* tracked a real baton using computer vision [4], and Ilmonen's *Virtual Orchestra* [5] is one of the few systems that also feature graphical output, synthetically rendering the orchestra as 3D characters. More recently, *You're the Conductor* [6] was designed to provide an immersive experience for children, and Nintendo's *Wii Music* [7] allows players to be conductors wielding a "Wiimote" instead of a baton.

However, there have been very few attempts to solve the opposite problem: an artificial (robotic) conductor conducting a real orchestra. There are several reasons behind this, one of them being applicability. While the entertainment and educational value of a system that allows even users with no prior experience to conduct orchestras is quite obvious, why would a human musician want to be conducted by a robot, and why would such a performance be interesting to the public?

One could envision several applications for a conducting robot. Perhaps the most obvious one would be education, where a robot could tirelessly conduct the same piece over and over again for the benefit of students learning either to play an instrument or to conduct. While robots might have a hard time improvising new gestures, they could ultimately be able to emulate different personal styles, switching effortlessly from Gustavo Dudamel to Leonard Bernstein and Herbert von Karajan. From a performance standpoint, an artificial system that could glean conducting information from sheet music would not replace a human conductor, but could augment his performance with features that entertain and educate the audience.

The other reason behind the lack of robotic conductors is technical difficulty. Of the few instances of robots conducting orchestras, perhaps the most known is Honda's ASIMO conducting the Detroit Symphony Orchestra performing "The Impossible Dream" in May 2008 [8]. While the event was clearly a success, some commentary afterwards focused on the fact that the robot's movements were pre-programmed and therefore strikingly un-musical. ASIMO, it turns out, could play back a recorded version of a conductor's movements, but lacked the ability to interact with the musicians. In fact, ASIMO was programmed to copy the gestures of the Detroit Symphony's education director as he conducted the same piece six months prior.

Other instances of conducting robots include Sony's QRIO leading the Tokyo Philharmonic Orchestra in a unique rendition of Beethoven's 5th Symphony in March 2004 [9], and Virginia Tech's humanoid robot DARwIn conducting the Roanoke Symphony in a short appearance at a 2008 Holiday concert [10].

All of these robots had two major flaws: they had to be reprogrammed for each musical piece, and they lacked the ability to provide feedback to musicians.

In this paper we address the first problem by introducing a family of systems that can conduct any musical piece in real time without having to be reprogrammed. Whether our artificial conductors are animated or robotic, humanoid or not, they rely on a parsing algorithm that we developed to read the score of a musical piece stored in a MIDI file and extract conducting information that allows them to generate and perform gestures that convey tempo, dynamics, and cueing to conduct a musical piece whose score is stored in a MIDI file.

2. SYSTEM OVERVIEW

Human conductors learn a new musical piece from its score. Since there is no single standard for the digital notation of sheet music, MIDI files have served as the ad-hoc standard. (Although newer music notation formats such as MusicXML [11] do exist, it has been suggested by [12] that the affordances of MIDI justify its continued use.) Therefore, we settled on using the MIDI format. Since the MIDI format was not intended for musical notation, it has its limitations, and we will discuss some of these throughout this paper.

The MIDI file format was originally designed to function as a serial protocol between different electronic instruments, and as such they can encode note and timing events, as well as volume information. Multiple channels originally designed to interface between multiple instruments can be used to encode the music to be played by different instruments in the orchestra.

A variety of software packages are available to generate MIDI files from sheet music or audio recordings. In addition, software packages can be used to manually annotate MIDI files with additional information that is not available in the notes themselves, such as articulation, dynamics, or even cueing. This annotation process would be akin to a conductor making notes on conventional paper scores.

In addition to knowledge of the score, conducting relies on one's understanding of gestures. While a general set of beat patterns does exist, and is documented in conducting textbooks, there is no general consensus regarding the allowed variations within gestures, or the exact meaning behind each gesture [13].

Given general knowledge of conducting gestures, and the MIDI file containing the musical score, our system will generate conducting gestures for indicating tempo, dynamics, and entrance cueing.

3. ALGORITHM

3.1. MIDI Parser

Much of the processing is done by the MIDI file parser, which needs to extract all the available musical information that is needed for conducting.

MIDI files will typically contain several tracks: one for each instrument in the orchestra, and an additional global track which contains information about the time signature, tempo, ensemble dynamic, and key signature.

Although the actual notes that have to be played are not important for conducting, all tracks of a MIDI file need to be analyzed from the beginning to the end. Events such as note-on, note-off, as well as changes in dynamics are stored in individual instrument tracks based on the time at which they occur. The global track contains information about global dynamic levels, as well as tempo and time signature, also labeled with their timing.

While humans refer to musical timing using beats, which are based on tempo, MIDI files use a unique metric of absolute time called ticks. Our parser converts from ticks to beats using the formulas:

$$seconds = \frac{60 * ticks}{tempo * conversionConstant} \quad (1)$$

and

$$beatsPerSecond = \frac{tempo}{60} \quad (2)$$

where the conversion constant is a value specified in the MIDI file.

The basis of the right hand gestures is formed by information about tempo and time signature in the global track. Left hand gestures include entrance and dynamics cueing.

Because cue data is not explicitly stored in MIDI files, the parser will have to use the note-on and note-off events from individual tracks to determine when each instrument is playing. An entrance cue is necessary when an instrument starts playing after a longer period of rest. However, human conductors don't have a set formula for how long (in number of seconds or number of measures) a rest period must be in order for the instrument to require a cue.

When determining the need for an entrance cue, our parser uses a set of thresholds that allow tempo to play a role in the number of measures that are considered a longer period of rest, loosely correlating the length to actual time.

Volume information is stored in all tracks: the global track contains information for the whole orchestra, while the other tracks have information for individual instruments. The parser analyzes all tracks looking for changes in dynamics, and generates requests for dynamic cueing gestures.

MIDI files encode dynamics using numerical values that are not always equivalent to the typical pianissimo through fortissimo notation, but since conducting only requires information about changes in dynamics, this did not impact system performance.

When no global or channel dynamic values are present, the parsing algorithm averages the stored dynamic levels for all instruments and stores the average as the global marking.

In addition to extracting information about conducting gestures, our algorithm helps musicians keep track of where they are in the song by calculating not only the relative beat number within the current measure, but also of the current measure number. This information is not stored explicitly in the MIDI file, but can be computed using formulas (1) and (2).

3.2. Gestures

The MIDI parser can be used to implement a variety of conducting systems, humanoid and non-humanoid robots, and animations.

For humanoid versions we used a Microsoft Kinect to capture the basic gestures of a human conductor. Unlike the method that was used to prepare for ASIMO's conducting performance, however, we did not capture the conducting of an entire musical piece. Instead, we collected geometric data about how the right hand gestures move during an individual beat pattern, and how the left hand goes up and down to indicate changes in dynamics. Our software used this data in conjunction with the current value of the tempo to provide gestures of appropriate speed.

Non-humanoid versions varied widely, but they all strived to display all the available information in a manner that is easy to follow by musicians.

3.3 Challenges, Solutions and Limitations

Due to the nature of MIDI files, our algorithm has some limitations.

MIDI files terminate when all instruments are done playing, rather than the song is intended to be over. That is, if a composition is intended to end with a rest interval, this will not be stored in the file. While this does not make any difference from an audio standpoint, it might make for an awkward ending to the conducting performance, leaving the conductor with the arms in the air. To address this problem, if the MIDI file ends before the end of the measure, the artificial conductor will perform additional gestures to conduct the measure to the end.

Since measures are not explicitly marked, possible rests at the end of a piece (which are fairly common) make pickup beats (notes that come before the first complete measure of a composition) virtually impossible to detect. To help with this situation, our program requires the user to manually indicate whether the composition has a pickup beat, and its length if one exists.

Another piece of information missing from MIDI files is articulation. Our current implementation therefore cannot handle it. An easy solution would be to use additional tracks in the MIDI file to manually annotate the composition with articulation information. Teaching an artificial conductor the fairly subtle differences between staccato and legato conducting gestures is also a challenging task.

More important limitations of an artificial conductor are the lack of emotion, improvisation, and feedback. Our students implementing the algorithm found the lack of emotion of a computer-based conductor to be especially troublesome and decided to address it for humanoid systems (in an admittedly very limited way) with a little trick. Artificial faces were designed to be able to convey a small number of emotions (happy, sad, excited, or neutral), and MIDI files were manually annotated with times during the composition where the conductor would have to convey these feelings. This solution generated its own problems, however: too much manual annotation would defeat the purpose of having a system that can automatically conduct any musical piece. And humanoid robots or animations with various facial expressions can get dangerously close to the "uncanny valley" [14], causing feelings of revulsion in musicians and audiences alike.

The lack of improvisation means that conducting performances of the same piece will always be identical, something that never happens with human conductors, but in an educational setting this might prove to be an advantage.

The lack of feedback is perhaps the most serious limitation of our current algorithm. Although all our conducting systems generated their gestures in real time, the lack of feedback to musicians made their performance no different from a prerecorded one. We plan to address this issue in the near future. We will start by investigating how an artificial conductor can ascertain tempo and volume information in real-time from the music that is being played and adjust its gestures to correct musicians if needed.

4. IMPLEMENTATION

The algorithm was implemented by undergraduate students at our institution in several artificial conducting systems including humanoid and non-humanoid animations, as well as humanoid robots. Four of these systems are shown in Figures 1-4 and detailed below. Videos of all conducting robots in action can be found on our project website at
http://www.tcnj.edu/~nakra/ConductingRobots.html

Link (shown in Figure 1) is a humanoid robot that conducts by moving its arms, turning on cue lights on its chest, and displaying different facial expressions on a screen that serves as its head. The robot was constructed by our students from scratch, using a galvanized steel frame. The arms, made of high density foam, are powered by Vex Robotics motors and controlled by Arduino microcontrollers, and have two degrees of freedom. The right arm keeps the tempo, while the left arm shows dynamic cues. Since implementation of entrance cueing would have required an additional number of degrees of freedom in the arms, and ideally a rotating torso, we opted for a different alternative: the chest features images of each instrument that light up to cue entrance. The face displayed on the monitor shows emotions and also helps with entry cues.

Roxy (shown in Figure 2) is a screen based humanoid conductor with a supplemental graphical interface. The conductor's right arm shows the beat pattern, while the left arm is used for entrance cueing. The

number in the top left corner of the screen shows the current measure, the bottom left corner displays the time signature. The graphical bar on the right represents the dynamic level, with low-fill for soft and high-fill for loud.

Figure 1. Conducting humanoid robot Link.

Figure 2. Humanoid conducting animation Roxy.

Carmen (shown in Figure 3) is a non-humanoid animation that represents conducting information through a variety of colored rectangles, with each color corresponding to a different instrument being played. The left side of the screen shows dynamics (with higher bars corresponding to louder music). The right side of the screen is used for cueing: bars drop down over the measure before the entry measure. The border flashes from white to black on the downbeat.

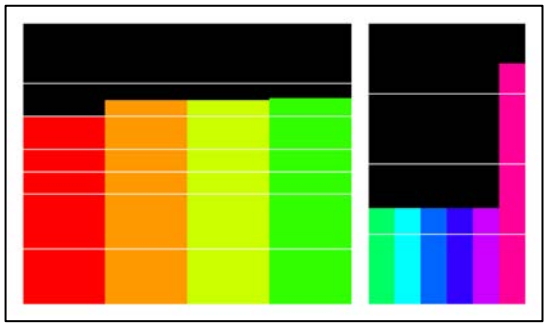

Figure 3. Non-humanoid conducting animation Carmen

Olmec (shown in Figure 4) is a hybrid humanoid/non-humanoid animation. It shows the current beat within the current measure as a dot circling around predefined positions. Entrance cues and dynamic change cues are encoded separately for each instrument through vertical bars on the bottom of the screen. An animation of a human face can issue nonverbal cues, such as changes in breathing, facial expressions, and nodding.

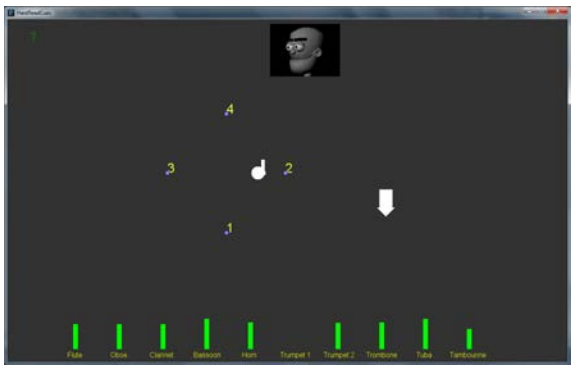

Figure 4. Non-humanoid conducting animation Olmec.

5. EVALUATION

A conductor, human or artificial, is useful if it can successfully convey information to musicians. To test the usefulness of our systems, and therefore the performance of our algorithm, we set up a live concert where our robots conducted a dectet of wind instruments played by undergraduate music majors at our institution. Each system conducted a different musical piece, rearranged for the dectet: Link conducted the theme from Dragon Roost Island in the *The Legend of Zelda* series of video games. Roxy conducted the song "For Good" from the Broadway musical *Wicked*. Carmen conducted the "Habanera" from Bizet's *Carmen.* Olmec conducted the song "Can't take my eyes off of you" by Frankie Valli. Figure 5 shows two of the conductors, Roxy and Olmec, in action.

At the end of the performance we surveyed musicians on their overall experience, as well as on reactions to individual robots.

Dectet members found being conducted by the four non-human conductors fairly acceptable overall, giving it an average rating of 7.22, with responses ranging from 4 to 10 on a 10-point scale anchored at *Unacceptable* and *Completely Acceptable*.

The musicians found the experience very interesting overall, giving it an average rating of 8.89, with responses ranging from 7 to 10 on a 10-point scale anchored at *Boring* and *Exciting*.

Musicians rated the artificial conductors only moderately effective overall at 6.78, with responses ranging from 5 to 9 on a 10-point scale anchored at *Not at all* and *Very*.

When asked to elaborate on their ratings, student musicians thought that being conducted by an artificial conductor is a fun experience, although the systems are

hard to work with, and miss some of the aspects of a human conductor.

Musicians also rated the conductor systems separately. All four conductor systems (listed in order below) averaged 6.9 or above on User Friendliness:

1. Olmec – 8.78
2. Roxy – 8.44
3. Carmen – 8.33
4. Link – 6.9

All four conductor systems (listed in order below) averaged 7.9 or above on Creativity.

1. Roxy – 9.00
2. Olmec – 8.67
3. Link – 8.22
4. Carmen – 7.9

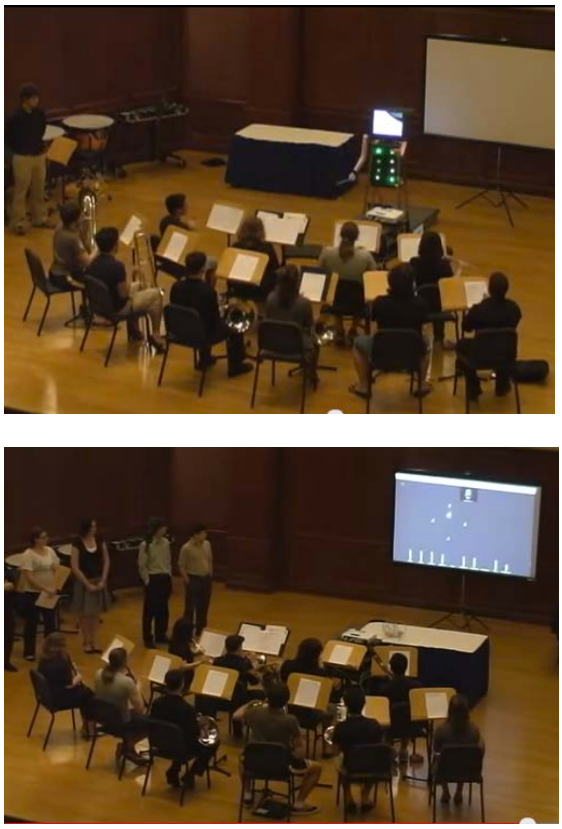

Figure 5. Two artificial conductors conducting a dectet of human musicians. Top: humanoid robot Link. Bottom: Non-humanoid animation Olmec.

6. CONCLUSIONS AND FUTURE WORK

In this paper we presented four artificial conductors, ranging from a humanoid robot through humanoid animation to non-humanoid animations, that conducted a small orchestra. Previous instances of conducting robots worked by prerecording the entire performance and requiring reprogramming for each new musical piece. Our systems use a parsing algorithm that we devised that allows them to conduct in real time any composition whose sheet music is available in MIDI format. The algorithm can extract information pertaining to tempo, dynamics, and entrance cueing, which can then be used to generate and perform appropriate conducting gestures.

All systems were successful in conducting a dectet composed of student musicians playing wind instruments. The musicians characterized the experience as very interesting overall (and admittedly their young age may have contributed to their openness). They found that the information conveyed by the artificial conductor was correct, and they were able to follow it with a little bit of practice. However, they rated the overall conducting performance as only moderately effective. This is understandable due to the lack of emotion and feedback of the system.

These artificial conductors represent an important first step towards robotic conductors that can have fully autonomous performances. Future work includes adding listening capabilities that would allow the system to provide real-time feedback to musicians, and the use of a different encoding for sheet music that would allow for easy representation of articulation and rest periods. The use of a commercially available humanoid robot would allow us to generate more human-like motions and concentrate on the musical aspects of the problem.

Acknowledgments

The authors would like to thank colleagues Chris Ault and Yunfeng Wang for their invaluable contributions to the robots and animations, as well as the students of the *Conducting Robots* class. This work has been supported in part by National Science Foundation grant #0855973.

7. REFERENCES

[1] M. V. Mathews, “The Conductor Program and Mechanical Baton,” *Current Directions in Computer Music Research*. MIT Press, Cambridge, 1991.

[2] T. Marrin, “Possibilies for the digital baton as a general-purpose gestural interface,” in *Proc. ACM CHI Conference on Human Factors in Computing Systems*, Atlanta, GA, 1997, pp 311–312.

[3] U. Satoshi and M. Yasunori, “A conducting recognition system on the model of musicians’ process,” *J. Acoustical Society of Japan*, 1998.

[4] D. Murphy, T. H. Andersen, and K. Jensen, “Conducting audio files via computer vision,” *Proc. of the Gesture Workshop*, Genova, 2003.

[5] T. Ilmonen, “The virtual orchestra performance,” in *Proc. ACM CHI Conference on Human Factors in Computing Systems*, The Hague, Netherlands, 2000.

[6] E. Lee, T.M. Nakra, and J. Borchers, “You’re the Conductor: A Realistic Interactive Conducting System for Children,” *Proc. NIME International*

Conference on New Interfaces for Musical Expression, Hamamatsu, Japan, 2004.

[7] Wii Music. http://www.wiimusic.com. *Last accessed 03/30/2014.*

[8] D.A. Durbin, "Honda Robot Conducts Detroit Symphony," *USA Today*, 5/14/2008. http://usatoday30.usatoday.com/tech/news/robotics/2008-05-14-asimo-robot-conductor_N.htm. *Last accessed 03/30/2014.*

[9] W. Knight, "Humanoid robot conducts Beethoven symphony," *New Scientist*, 04/04/2004. http://www.newscientist.com/article/dn4845-humanoid-robot-conducts-beethoven-symphony.html. *Last accessed 03/30/2014.*

[10] "DARwIn Conducting the Roanoke Symphony". https://www.youtube.com/watch?v=WmriV8Y0kcE. *Last accessed 03/30/2014.*

[11] MusicXML for Exchanging Digital Sheet Music. http://www.musicxml.com/. *Last accessed 03/31/2014.*

[12] S. Cunningham, "Suitability of musicxml as a format for computer music notation and interchange." *Proc. IADIS Applied Computing International Conference*, Lisbon, Portugal. 2004.

[13] R. Dannenberg, D. Siewiorek, N. Zahler, "Exploring Meaning and Intention in Music Conducting," in *Proc. Int. Computer Music Conference*, New York, 2010.

[14] M. Mori, "The Uncanny Valley," translated by K. F. MacDorman and T. Minato, in *Energy*, 7(4), pp. 33-35. *Last accessed on 03/31/2014 at* http://www.movingimages.info/digitalmedia/wp-content/uploads/2010/06/MorUnc.pdf