Gesture-Driven Interaction Using the Leap Motion to Conduct a 3D Particle System: Evaluation and Analysis of an Orchestral Performance

Joyce Horn Fonteles
PPGIA
Universidade de Fortaleza (UNIFOR)
60811-905 Fortaleza–CE Brasil
hornjoyce@gmail.com

Édimo Sousa Silva
PPGIA
Universidade de Fortaleza (UNIFOR)
60811-905 Fortaleza–CE Brasil
edimossilva@gmail.com

Maria Andréia Formico Rodrigues
PPGIA
Universidade de Fortaleza (UNIFOR)
60811-905 Fortaleza–CE Brasil
andreia.formico@gmail.com

***Abstract*—In this work, we present and evaluate an interactive simulation of 3D particles conducted by the *Leap Motion*, for an orchestral arrangement. A real-time visual feedback during gesture entry is generated for the conductor and the audience, through a set of particle emitters displayed on the screen and the path traced by the captured gesture. We use two types of data input: the captured left and right hand conducting gestures (some universal movements, such as the beat patterns for the most common time signatures, the indication of a specific section of the orchestra, and the cutoff gestures), which are responsible for setting the *tempo* and dynamics of the orchestra; and a MIDI file, which contains information about the score of an orchestral arrangement, defining which notes each musical instrument should play. As regards the gestural input of the conductor, we have considered two musical elements of expression: *tempo* and dynamics. Besides performing functional testing, we analyzed the simulation results focusing on the occurrence of false positives, false negatives, positional deviations and latency. Moreover, a professional conductor evaluated our system and provided qualitative feedback about it.**

Keywords – Gestural Interaction, 3D Particles, Orchestral Arrangement, Visualization, Performance Analysis, Evaluation, *Leap Motion*.

I. Introduction

The evolution of motion tracking technologies has allowed for the development of new and innovative human-computer interfaces that help to improve user experience, bringing new possibilities of computer applications that fit more accurately to their real-life counterparts.

In spite of the vast amount of existing work, traditionally based on techniques of Computer Vision, gestural recognition of hands and fingers is still far from satisfactory in real-life applications [22], mainly because most of the algorithms are dependent on ambient lighting. Recently, this problem was solved using new devices that have infrared cameras, like the *Leap Motion Controller* [6]. These devices also offer features for image segmentation (due to their depth sensors), enabling the development of interactive, gesture-controlled applications at a considerably lower cost, without degrading accuracy (or resolution) [10]. The *Leap Motion* sensor also offers additional advantages: it is a small device, easy to set up, very precise, and does not suffer significant performance loss due to problems resulting from occlusion, thus making tracking technology more accessible to the average consumer.

Gestural interfaces have already proved to be an essential component for interactive applications, with great potential to provide an experience in virtual reality which more closely resembles that of the real world [15]. They have provided highly significant technological advances in terms of robustness, speed and tracking precision. More specifically, the tracking process is done by sensors that extract and recognize gestural patterns via inputs of the *data stream* type. This process has been described as a complex task due to the discontinuities of the captured coordinates. Consequently, the design success of an application of this kind lies in the way the gesture recognizer's state changes over time, correlating this information to the system requirements and client feedback [21].

Currently, interactive virtual applications with gestural control have gained visibility, improving the way people interact with computers [25]. In the musical area, the *Leap Motion Controller* has opened up new possibilities for mapping gestural controls for musical expression, performed by the human hand [12].

Despite the benefits generated with the use of these devices, the correct identification of the user's gestures is still challenging: the same gesture can be done at different time intervals and represented in many forms and styles, varying from person to person [13]. Basically, these issues are associated with handling gestures (free trajectories, which often require continuous feedback with difficulties related to efficient tracking of a particular member, such as the user's hand); and command gestures (predefined trajectories with issues related to pattern matching) [2]. The identification of a gesture begins with tracking the member of interest, followed by the treatment and processing of the captured data. Usually, this procedure is done by *Support Vector Machines* (SVM) [1], associated with the training of neural networks for the identification of gestural patterns, there being several variations of this approach [20], [3]. However, the use of a SVM is part of a complex process,

ISSN: 2236-3297

with a high learning curve.

The challenges regarding the visualization of a musical structure reside in the quality of the captured input gesture data and of the graphical design representation for the musical piece. The visual feedback provides concrete information about the ways in which the user can control the application through gestures and how effective is this control. However, neither the potential to display the musical structure of a piece and its visual elements [17], nor different forms of orchestral conduction via gestural interface (in which a set of musical instruments is organized into sections) have been fully explored. In the past, several initiatives focused on virtual forms of conducting using the computer, but they were not promising concerning the level of user satisfaction or the quality of the perceived user experience.

We present and evaluate an interactive simulation of 3D particles conducted by the *Leap Motion*, for an orchestral arrangement. A real-time visual feedback during gesture entry is generated for the conductor and the audience, through a set of particle emitters displayed on the screen representing the sections of an orchestra, and the path traced by the captured gesture. This work extends previous publication [9], adding new functionalities for the identification of gestures, increasing the possibilities of interaction between the conductor and the system using the left hand to provide specific cues and dynamics; and both hands to do cutoffs on the orchestral performance. We also mapped a new musical variable, *i.e.*, transparency, to the particle's visual attribute with the aim of representing the strength of each beat on a measure, according to the executed beat pattern. Differently from [9], the trajectory of the conductor's gesture is currently displayed using a Catmull-Rom spline [5] to represent visually the fluidity of the executed musical movements in a better way. Additionally, we have also performed testings focused on new compound time signatures to analyze how the three beat patterns (binary, ternary, and quaternary) behave in a more complex conducting experience, considering that each beat of a measure is divided into three equal parts, which means that each movement of a gesture has a triple beat instead of just one (as occur in the simple time signatures). Moreover, a professional conductor was invited to evaluate and to provide qualitative feedback about our system.

Basically, our system aims to facilitate the understanding of musical compositions for an ensemble of instruments, while emphasizing the importance of the conductor for a performance of a musical piece, through the visual mapping of abstract sound elements (timbre or tone color, volume or sound intensity, and pitch), conducting patterns and music notation scores. We use two types of input data: the conductor's captured gestures (some universal movements, such as the beat patterns for the most common time signatures, the indication of a specific section of the orchestra, and the cutoff gestures); and a MIDI file containing information about the score of an orchestral arrangement (in this work, for the song *Bohemian Rhapsody* by *Queen*). Regarding the conductor's gestural input, we have considered two elements of musical expression: *tempo* and dynamics. Besides performing functional testing, we analyzed the simulation results focusing on positional deviations, latency, and the occurrence of false positives and false negatives with regard to the beat patterns performed with the right hand, and also the selection of a specific section of the orchestra with the left hand.

II. Related Work

Gestural interfaces depend on the computer to identify a pattern in space for recognizing human body movements, such as of the user's fingers, hands or whole body, and for translating these movements into specific actions, as part of the software component interface of the application [15]. Once captured, the sequence of gestures must be analyzed in real time using specific algorithms to determine when a particular movement has occurred and to which pattern class it belongs.

So far, visual gesture feedback has not been widely explored, especially the continuous type [14]. Some techniques subdivide a gesture into several less complex ordered sub-gestures [18], as an attempt to solve this problem. The process is divided into the following two steps: identification and pattern matching of the sub-gestures. If there is a similarity between the captured and the stored sub-gestures, they are considered as successfully identified. A downside of this approach is that the success rate in identifying gestures is directly dependable on the quality obtained in the system training phase.

The *Leap Motion Controller* provides a non-invasive method of independently tracking the captured data, related to the user's hands and fingers. Before the launch of this device in mid-2013, no other commercial sensor offered this level of precision [12].

To the best of authors' knowledge, the most popular application for gestural control of music software is *Geco* [4]. Basically, MIDI messages (sound effects of mixing and volume) are mapped through a gestural interface, limited only to commands using closed or opened hands. Besides being commercial, another disadvantage of this software relates to the control of musical parameters, which is not done through continuous tracking of the hands and fingers.

Some very recent published works in the musical area using the *Leap Motion* for isolated control of digital instruments, commonly the piano, have been proposed [17], [12], [11], [24]. For instance, in [24], the *Leap Motion* device is used in conjunction with a "glass board", containing markings of a piano's keyboard made with ink. The results show that when performing movements with greater speed, false positives and negatives are generated. In [11], the *Leap Motion* is used as an electronic music mixer in real time. The gestures are activated to generate several equalizer effects. The gestural control of sets of virtual musical instruments in an orchestra, arranged together and/or in an independent form, is indeed an even more challenging and complex problem.

We believe that initiatives like these, can contribute to the creation of a computer support tool in conducting, since a lot of practice time is required to be a conductor. Besides, it is almost impossible to gather more than 200 orchestra members for daily practice. To learn about *tempo*, instrumentation, and melody line, conducting students using MP3 files usually practice with music whose speed is unchangeable. In addition, as much as possible, they study with pianos to make up for what they lack in live music practice, having to quickly translate the full score of the orchestra to the piano music. Therefore, the conductor cannot expect to have a high quality level of interaction with an orchestra in a short time, as one would expect with an individual musical instrument.

III. The Leap Motion Device

The *Leap Motion Controller* (Figure 1) is a compact device developed by *Leap Motion* for gestural control [7]. It is portable and has a brushed aluminum body with a black glass on its top surface that hides two CMOS sensors and three infrared LEDs, which work together to track hands and fingers in interactive applications. The positions of the hands and fingertips are detected in coordinates relative to the center of the controller, taking as reference, the right-handed coordinate system. Several gestures can be natively identified by the *Leap Motion*. Another positive feature is its affordability. It costs approximately US$ 80, which collaborates to its popularity.

Fig. 1: The *Leap Motion* device.

The computer control is done using its three emitters and two infrared cameras, at a frequency of 290 frames per second. The *Leap Motion* captures gestural information, and also identifies the main hand joints. In comparison with other devices for gestural identification, currently available in the market, such as *Kinect, Playstation Move, Nintendo Wii*, etc., the *Leap Motion Controller* has an advantage, since it is the only device able to identify a native core set of gestures. However, it does not capture sound or color images. According to the manufacturer, the sensors accuracy for position detection is approximately of $0.1mm$.

The *Leap Motion*'s field of view has a format similar to an inverted pyramid, whose lower area length measures $25mm$ and the top one $600mm$, with 150º of field of view, as shown in (a) of Figure 2. The gestural tracking precision is inversely proportional to the distance between the device and the user's hands. Consequently, for accurate identification of the hands, ideally, it is expected that they are positioned at a height which varies between 10 to $20cm$, with the *Leap Motion* positioned between the keyboard and the user, as shown in (b) of Figure 2. It is also important to ensure that the device's lens are spotless, as these can directly influence the performance, and that the actions are performed within the device's field of view.

The *Leap Motion SDK* (available for *C++, Java, Objective-C, C#, Python, Javascript,* etc.) can be used to develop applications that exploit the capabilities of this device, compatible with Windows and OS X operating systems. Currently, it provides high-level functions, such as [23]: (1) detection of the hands and their positions in space, according to the field of view; (2) hands' orientation and palm-bending; (3) geometric transformations, such as, scale, rotation and translation, from the hands' movement; and (4) 3D orientation and individual positioning of the fingers, as well as normalized 2D positions in the screen space.

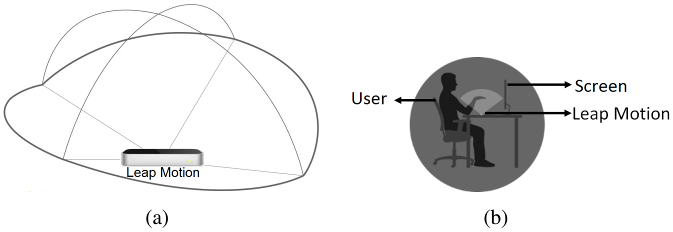

Fig. 2: In (a), the *Leap Motion*'s sensor area; and, in (b), the *Leap Motion* positioned between the keyboard and the user.

Obviously, the change of habit in using a new input device causes new challenges, which are followed by an adaptation phase. This would be no different with the *Leap Motion*, stimulating a new learning curve. The device also presents a small limitation related to its hardware: movements that cause the overlapping of fingers, for instance, an upright pincer movement, are difficult to be recognized requiring the user to get adapted to this limitation and, whenever possible, to avoid this type of gesture.

IV. System Overview

A. Description

Just as athletes practice thousands of times before competing, conductors must also practice conducting the basic patterns, and much of it without conducting a choir or orchestra. One must practice until the patterns become automatic as well as easily conducted with metric accuracy, while also possibly doing something else. The ability to reproduce the patterns physically is easily within the grasp of any average person with normal coordination. Interpretation of the music, however, lies far beyond the mere reproduction of the beat patterns. A good technique is of value only to those who study and understand the musical score, who establish a good rapport with their ensemble, and are able to transmit the ultimate beauty of music with sensitivity.

In real life, with the movements of the right hand, the conductor sets the beat for the musicians and the *tempo* with which the musical piece has to be performed. Otherwise, each musician would play their respective musical instrument in their own way. As the conducting itself is transmitted essentially by the hands, usually, the right hand marks the beats through a set of patterned gestures, while the left one indicates the entries and dynamic, always seeking to obtain a colorful rich sound during the performance of a musical piece.

Thus, in the context of this work, conducting is the art of transmitting to a set of virtual particles emitters, a rhythmic and expressive content of an orchestral arrangement of the musical piece *Bohemian Rhapsody*, by *Queen*. We have implemented three different functionalities to simulate the experience of orchestral conducting.

More specifically, the first one focuses on the right hand, the conductor's main tool to indicate the beats, where three of the most fundamental patterns in Western Music, the two-beat, three-beat, and four-beat are recognized. The beat is similar to the "pulse" of the music, with logical divisions.

The second functionality focuses on the conductor's left hand. Since the left hand is basically free from time-beating chores, it can be used for cueing, to indicate dynamics and style to a specific section of the orchestra. Cueing refers to the numerous times a conductor needs to indicate important entrances (while the music is in progress) or important parts that need to be emphasized. One of the utmost importance to a conductor is the practice of giving cues, so that the two arms operate independently of each other. The beat pattern with the right hand must continue unhindered as the left hand executes an entirely different type of function.

The third functionally is the cutoff, the conductor's gesture to indicate that the musicians should stop producing sound, which is performed with both hands. Many musical pieces end with a *fermata*, a held note, and it's up to the conductor to define the release or cutoff after the hold. For the conductor, the cutoff indicates a sense of finality, whether it is the last beat of the piece or a beat that needs a clearly defined cut during the performance.

A visual and gestural feedback is generated for the conductor and the audience, through the display of a set of 3D particle emitters and the gestures trajectory captured with the *Leap Motion* (Figure 3). There are two types of input data into our system: the conductor's gestural capture, responsible for setting the *tempo* and the orchestral dynamics; and a MIDI file, that defines which sounds each instrument must play. Regarding the conductor's gestural input, we consider two musical elements of expression: *tempo* and dynamics. The timing of the beats of a song is typically indicated with the right hand of the ruler. Hand draws a shape in the air for each measure depending on the time signature, by indicating each beat through the change of movement in different directions. Thus, the speed with which the ruler performs the movements of each beat pattern implies the execution speed of music and computer animation.

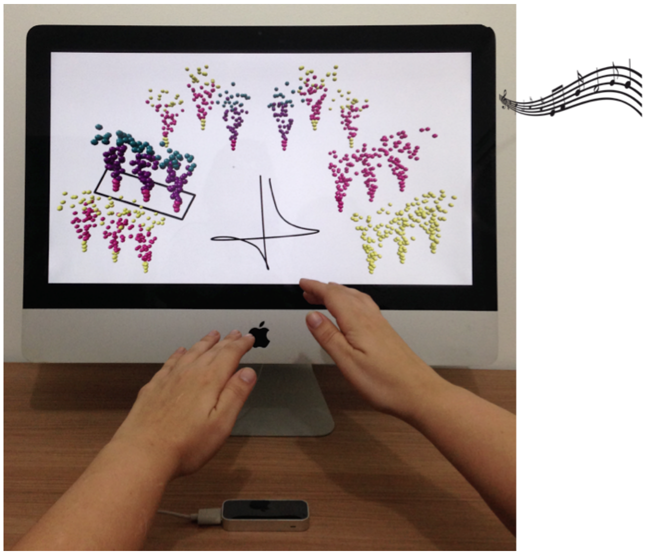

Fig. 3: Illustrative example of a user using the *Leap Motion Controller* to conduct the particle emitters, in an orchestral arrangement modeled for this work.

The dynamics of a musical performance is commonly associated with the amplitude of the conducting gesture, in which larger and wider shapes represent stronger sounds. Thus, when the conductor performs a broader movement than expected, the music volume increases and so does the particle size in our application.

We also mapped three sound variables as visual elements of the particles: timbre, volume and pitch. We modeled the timbre, also known as tone quality, as a particle emitter. It represents the sound quality, indicating which instrument performs a specific sound. For each family (or subdivision of instruments of the same family), a group of emitters was generated and positioned in accordance with the standard distribution of instruments in the seating arrangement for an orchestra. We mapped the volume or loudness, as the size of the particles, which is a sound quality indicating the strength with which a sound is executed. In our work, more intense sounds imply growth within the sphere representing a particle. Finally, we mapped the pitch, which indicates what is the musical note of a sound, to set the particle colors of each emitters' set of the same family (or subdivision). Figure 4 shows the arrangement of the mapping we defined to the particle emitters distributed in sections of an orchestra.

The most basic function of the conductor is keeping the whole orchestra coordinated, so that all musicians start and stop playing at the right times. The key to this is to express the beat of the music through hand gestures, but even simple beat patterns are not always easy to follow. Since the musician's perception of the beats within the conductor's gestures is crucial for a well executed musical piece, it became necessary to take into account a more appropriate form of visually representing these beat patterns, while also trying to identify and better understand the main difficulties in recognizing the correct time of a beat within certain conducting gestures. Many conductors' manuals suggest that the beat is indicated when the conductor's motion changes direction, while the hand is moving at the highest speed. However, basic laws of Physics would suggest that a direction-change is accompanied by low speed. In our solution, we considered the following: if a conductor's motion actually traces a wide loop, consequently, it does not have to change speed as it changes direction. Studies have shown [16] that the arc width created by the conductor's movement does not impact on the perceived beats, but the velocity and acceleration along the motion path of the conductor's hand do. We used Catmull-Rom splines [5] to represent visually the gestures trajectory, that is, the specified curves will pass through all of the control points captured by the *Leap Motion Controller*.

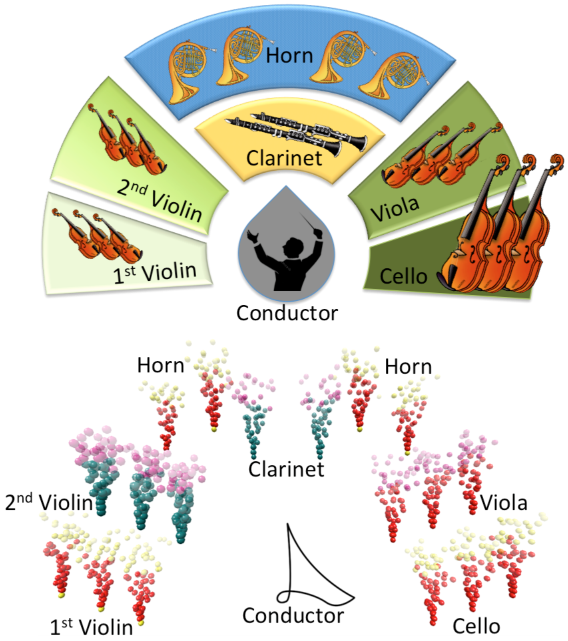

Fig. 4: Orchestral seating arrangement and distribution of section/instruments: string in shades of green (3 first Violins, 3 second Violins, 3 Violas, and 3 Cellos), brass in yellow (4 Horns), and woods in orange (2 Clarinets).

B. Gesture Recognition Regions

In this work, the interaction with the orchestral instruments is performed through gestures made with the right hand, and/or the left hand.

For the right hand, we have implemented three specific conducting gestures for music interpretation: two-beat or binary, three-beat or ternary, and four-beat or quaternary [8]. Thus, we map the identification area of the conducting gestures captured by the *Leap Motion* in the form of 10 regions, as shown in Figure 5.

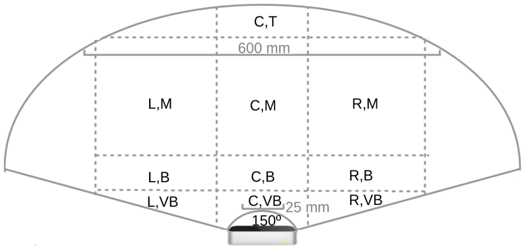

Fig. 5: Regions of gesture recognition: (C, T) = (Center, Top); (L, M) = (Left, Middle); (C, M) = (Center, Middle); (R, M) = (Right, Middle); (L, B) = (Left, Bottom); (C, B) = (Center, Bottom); (R, B) = (Right, Bottom); (L, VB) = (Left, Very Bottom); (C, VB) = (Center, Very Bottom); (R, VB) = (Right, Very Bottom).

After mapping the recognition area, the region in which the right hand of the user is positioned is identified. While performing the gestures, the right hand moves around several of the mapped regions. Due to the fact that the hand movements are tracked continuously by the *Leap Motion* device, each region where the hand dynamically passes through is stored in our implementation as a component of an *ArrayList*. The system only starts recording the hand positions when it leaves the region (C, T). As soon as the hand returns to the region (C, T), the gesture is identified, since all the modeled conducting gestures begin and end in this particular region. The next step is to traverse the array, searching for the sequence of regions that are compatible with one of the three well-known conducting gestures: binary, ternary and quaternary, shown in Figures 6, 7 and 8, respectively.

In particular, all the three modeled gestures have two positional characteristics in common: they start in the region (C, T) towards the region (C, M), and end in the region (R, B) or (R, M), towards the region (C, B).

The gesture recognition process is done as follows. As regards the binary gesture, we divided the device's field of view into 9 regions through which the right hand of the user should pass consecutively (Figure 6). With regard to the ternary and quaternary gestures, we used 8 and 11 regions, as shown in Figures 7 and 8, respectively.

Due to the complexity of the quaternary gesture, it may be possible that areas located beyond the specified pattern list that represents this gesture are included in the array during hand's movement, therefore, adding some sort of noise that must be excluded from the data structure.

For the left hand, to simulate the conductor selecting a specific group of instruments, the conductor hand must be placed in a neutral position, and then moved towards a particular direction. Each direction to which the conductor points at is mapped to a different section of the orchestra.

To identify the gesture of pushing the left hand in a certain direction, we used a vector-based approach towards the identification. The user positions the hand in a neutral position which is represented by the vector $< 0, 0, 0 >$. From the point

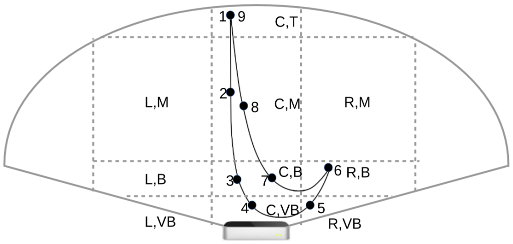

Fig. 6: Sequential positions of the user's hand, tracked by the *Leap Motion*: (1-C, T), (2-C,M), (3-C, B), (4-C, VB), (5-R, VB), (6-R, B), (7-C, B), (8-C, M) e (9-C,T).

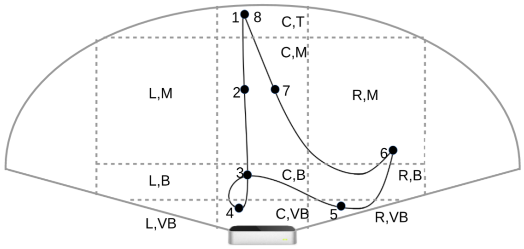

Fig. 7: Sequential positions of the user's hand, tracked by the *Leap Motion*: (1-C, T), (2-C, M), (3-C, B), (4-C, VB), (5-R, VB), (6-R, M), (7-C, M) e (8-C, T).

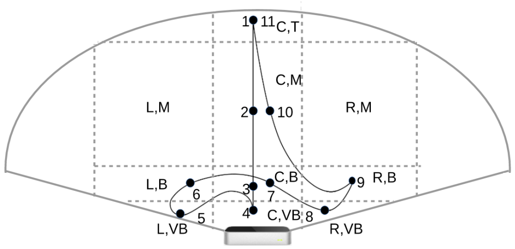

Fig. 8: Sequential positions of the user's hand, tracked by the *Leap Motion*: (1-C, T), (2-C, M), (3-C, B), (4-C, VB), (5-L, VB), (6-L, B), (7-C, B), (8-R, VB), (9-R, B), (10-C, M), (11-C, T).

of view of the user, this point is located on the upper left corner. The user should then push his/her hand until a distance of $4cm$ is found between the neutral position and the actual desired position. When this occurs, we consider the current position as the final position and create a vector representing this movement from the neutral position to the final one. We created the following six vectors $< -1, -1, 1 >, < 0, -1, 1 >, < 1, -1, 1 >, < -1, 1, 1 >, < 0, 1, 1 >, < 1, 1, 1 >$ to represent the possible movement directions of the left hand, as shown in Figure 9. The angles between the vector that represents motion and those that represent the instrument groups are calculated. The lowest measured angle indicates which set of instruments has been selected.

To represent the conductor's cutoff gesture, both hands must be closed. As previously mentioned, the *Leap Motion SDK* provides several gestures that can be natively identified, such as counting how many fingers are visible on a hand. Thus, making necessary just to verify if there are no visible fingers on both hands. In real life, a conductor can give a cutoff using either or both hands. Our implemented solution tests both hands to avoid false positives, while giving a cue with the left hand or gesturing a beat pattern with the right one.

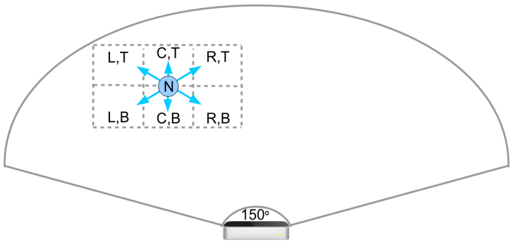

Fig. 9: Possible movement directions of the left hand: L,T (left, top), L,B (left, bottom), C,T (center,top), C,B (center,bottom), R,T (right,top), R,B (right,bottom).

C. *Implementation Details*

For the development of the application, *Java* language was used due to its compatibility with some libraries available for the *Leap Motion*. To access the information provided by the device, we used the *Leap Motion SDK v1.2* [7], which provides the information captured by the *Leap Motion* in a simple, objective and well documented manner, which facilitates software development and minimizes the need of implementing intermediate layers.

The user interface has been adapted to the screen resolution, in such a way as to map the coordinates of the *Leap Motion Controller* (mm), in screen coordinates (pixels). Since the *Leap Motion* device has a high hand tracking precision, it was necessary to perform a calibration adjustment in tracking. To reduce sampling rate (the number of data points captured), we used as reference the resting hand, so as not to generate large variations in time. Otherwise, it would be necessary to treat a redundant and unrepresentative number of positions.

Java was also used for the recognition of gestures performed in the device's field of view. The *Leap Motion Controller* captures a frame, at any given point in time. Our application accesses this data via the *Leap Motion* API. In particular, a *LeapHand* type object contains an identifier, properties that represent the physical characteristics of the hand, including its position in the Cartesian plane, which is the most relevant information for the identification of gestures.

The interaction between the system components is shown in Figure 10. As input, the system receives two types of information: gestural controls captured with the *Leap Motion* and MIDI files. Its three structural components are *Java Virtual Machine* (JVM), *Leap Motion SDK* and *OpenGL*, all three currently accessible through the major existing operating systems. The core of the system consists of four components that constantly exchange information: (1) *Gesture Identifier*, (2) *Particle Manager*, (3) *Synchronizer*, and (4) *Renderer*.

The *Gesture Identifier* recognizes gestures and sends messages to the *Particle Manager*, through the *Synchronizer*. The *Synchronizer* is responsible for communication amongst the other three system components. Upon receiving the messages, the *Particle Manager* changes the execution of the MIDI files and sends information to the *Renderer*, so that it can display the animation on the computer screen.

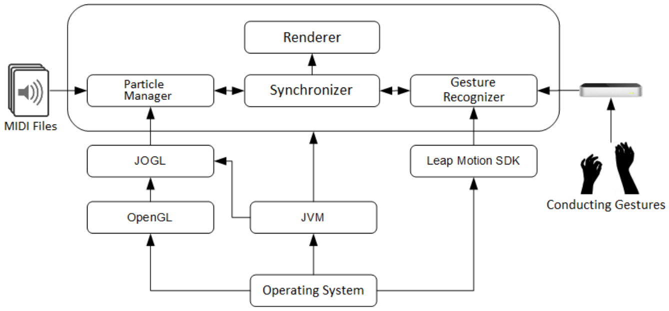

Fig. 10: System components.

V. Evaluation

A. Specialist's Impressions

A female conductor with 28 years of experience was invited to test our system, evaluate it and give some insights. She had never heard about or used the *Leap Motion* device before.

Initially, to simulate the steps taken by a typical conducting student, an MP3 version of the song *Bohemian Rhapsody* by *Queen* was played to familiarize the conductor with the musical piece. The conductor spent a few minutes studying the score, taking notes of the changes in time-signature, *tempo*, and dynamics.

After a brief explanation of the system's main functionalities, a first test run was executed, where the conductor simply tried to make the beat pattern gestures according to the time-signatures on the score, while testing how wide the conducting movements could be, in order to be successfully captured by the *Leap Motion Controller*. A second test run was then executed, focusing only on the selection of different sections of the orchestra, following the melodic lines of the song and doing cutoffs. Finally, the third and final test run was performed by the conductor using both hands, focusing on two important elements of musical expression: *tempo* and dynamics.

The conductor was very pleased using the *Leap Motion Controller*. During the first two training sessions, she noticed that when using both hands at the same time, wide gestures lost tracking. On her own initiative, looking at the animation of particles displayed on the computer screen, she was able to calibrate the amplitude of her gestures to fit in the device's tracking area. The conductor also gave a positive feedback about the visual mapping of pitch, timbre, and volume, commenting that it is quite common for musicians the association of pitch and color, and of volume and size.

In the very beginning, during the adaptation phase in using a totally new input device and computer application, the particle emitters seemed to cause some visual confusion to the conductor. However, as soon as the conductor observed more carefully the color pattern, she easily identified which ones represented each section of the orchestra. The conductor was even more convinced and satisfied when she used the left hand to select specific sections.

Overall, the conductor impressions were positive, although she also made some observations about the importance of including other expressions in the application, such as articulations, which she believes that are also very important to a conductor. This observation will contribute to improve the design of our system and should be implemented in the future as a new system functionality. The conductor also questioned the developers about how a percussion instrument might be represented on the computer screen, since some of them do not have a defined pitch attribute on the score.

B. Tests and Results

Functional tests were also conducted to perform quantitative analysis. The execution platform was an iMac (OS X version Yosemite 10.10.2), processor 2.5 GHz Intel Core i5, 4GB Memory 1333MHz DDR3, and graphics card AMD Radeon HD 6750M 512MB. As already mentioned, an orchestral arrangement for the song *Bohemian Rhapsody* was created for six groups of instruments, focusing on the three conducting beat patterns previously described. The *Leap Motion* device was placed between the keyboard and the user, with the user's hands at a height of 10 to $20cm$, calculated from the base of the table.

A major challenge for anyone aiming to understand, comprehend and appreciate orchestral music is that its score is not simple to understand and study, since the symbols may represent different notes, according to the instrument for which they are written. Among the instruments we have chosen to represent the string section (Violins, Violas and Cellos), we can observe that each one of them has a different clef. A clef is a symbol placed on one of the lines at the beginning of the stave to indicate which note is on that line, thus serving as a reference point from which the notes on any other line or space of the stave may be determined [19]. Violins are written in the treble clef or G-clef, Violas use the alto clef or C-clef, and Cellos use the bass clef or F-clef. This situation is shown in the tests performed on the section highlighted in Figure 11: The 2nd Violins and Cellos have the same marking on the stave, however, they emit different notes because their different clefs, which makes the 2nd Violins to emit yellow particles, and the Cellos, purple ones.

In addition, the instruments chosen to represent the wood and brass sections, respectively, the Clarinet in B♭ and the Horns in F, present an even greater challenge (even for professional musicians): they are transposing instruments. It means, for example, that when performing a C note, the sound produced will not be of this note because it will be transposed according to the characteristic of each one of these instruments. During the simulation tests shown in Figure 11, the selected frame of the animation illustrates the following instruments: 2nd Violins and Horns in F, with transmitters emitting particles of the same color, and emitting the same sound, despite their notes in the score being different, and the clef, the same.

The decision to map the pitch of the musical notes as particle colors was motivated by these challenging features, since it is much simpler to identify that different instruments are emitting the same sound by seeing that their particles have the same color, than by analyzing the notes written on the score, with their different clefs, and possibilities of transposition.

In music, dynamics usually refers to the loudness of the notes. These dynamic indications on the score are relative, not absolute, since they do not indicate an exact level of volume (just that a note or phrase should be louder or quieter that the previous executed notes). Interpretations of dynamic levels are left mostly to the conductor. Therefore, it is the conductor's job to set the shape of the sounds produced by the instruments played by the musicians of the orchestra, taking into account the dynamic instruction set by the composer in the score, and the personal desire to create a different interpretation on specific musical moments of the performance.

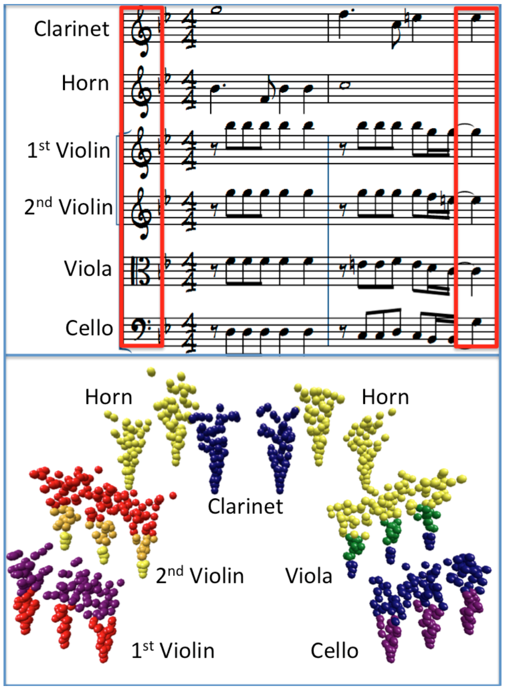

Fig. 11: Animation of the particle emitters, controlled by the *Leap Motion*: mapping of pitch from MIDI.

The dynamics is commonly associated with the amplitude of the conductor's movement, in which larger and wider shapes represent stronger sounds, requiring the musicians to observe the movements that the conductor performs, so that they can follow it accordingly.

In Figure 12, there are two images that capture the same musical moment, however, during two different simulation test runs: the difference in particle size displayed at the top and bottom of Figure 12 is directly related to the amplitude of the quaternary compass movement performed by the conductor, captured by the *Leap Motion* and mapped on the computer screen. The second test run differs from the first one, since the conductor aimed at conducting the orchestra by generating a louder sound, increasing the size of the conducting pattern. Consequently, the particles behaved exactly like a musician, following the conductor's instructions, thereby increasing their sizes. It is also important to note that the emitters representing the 2nd Violin in both images of Figure 12 produce particles larger than those of other emitters. This is because there is a dynamic attribute in the 2nd Violin score for this piece of music. Thus, both the dynamic indication input of the MIDI, and the conductor's gestural input are obeyed, showing that they are not mutually exclusive.

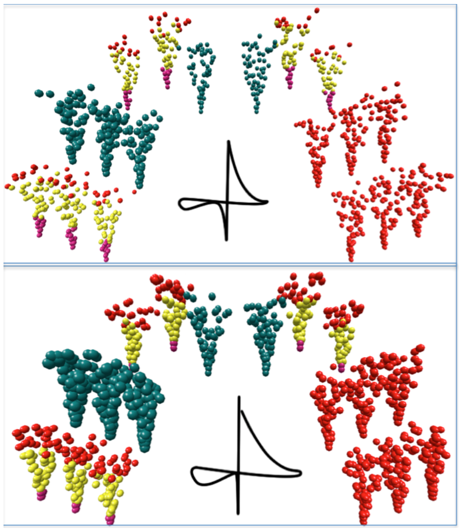

Fig. 12: Animation of the particle emitters, controlled by the *Leap Motion*: amplitude of the conductor's movement and the associated dynamics.

For the right hand tests, regarding false positives (FP), false negatives (FN), positional and latency deviations, three different segments of the piece were analyzed, each one composed of six measures.

In particular, the song is highly unusual for a popular piece in that it features no chorus, and the musical format of writing changes in style, tone and *tempo* throughout the execution. The frequent shifts in *tempo* and in rhythmic character, from one section of a composition to the next, proved to be a challenge because this song has two types of time signatures: Simple Time and Compound Time. The Simple Time signatures are 2/4, 3/4, and 4/4, that is, respectively, the binary, ternary, and quaternary beat patterns mapped by the *Leap Motion*. In these fractions, the numerators indicate how many beats exist per measure, and the denominator equals to 4 means that a quarter note is the unit of measurement, representing one beat. The Compound Time signatures are 6/8 and 12/8. The denominator equals to 8 indicates that an eighth note is the unit of measurement. In essence, the 6/8 and the 12/8 are compound forms of the 2/4 and 4/4 patterns, respectively, which means that when a song has a fast *tempo*, like the *Bohemian Rhapsody*, the number of beats is far too many to be represented as individual gestures. Thus, the patterns for the simple time signatures should be used [16], that is, each movement on a pattern will represent more than one beat.

Therefore, the first selected segment is the intro of the song composed of six bars: the first two are four-beat patterns, followed by one bar in two-beat, one bar in three-beat, and another two bars in four-beat. The second segment was chosen from the beginning of the operatic section, because while the underlying pulse of the song is maintained, the dynamics varies greatly from bar to bar, with the first three bars in four-beat, followed by one bar in two-beat, and ending with two bars in four-beat. The third segment was selected to verify how the particle system would behave with compound time signatures, with the first three bars in 12/8, conducting with four-beat pattern, followed by one bar in 6/8 with two-beat, and ending with two bars 12/8 in four-beat.

To identify the success rate in the captured gestures, each of the three sections of the song was performed 10 times. The simulation tests were done with the aid of an electronic metronome (a regular device that produces pulses of duration), to maintain a constant *tempo*, regardless of the gestures performed on a metric of 72 beats per minute. We found three distinct situations for the right hand, along the orchestral performance: successfully recognized gesture, False Positive (FP) and False Negative (FN), as shown in Table I.

Every time a gesture was successfully recognized, the particle system and the system's audio output reacted as we have expected, regarding the dynamics and *tempo*. The occurrences of FP, in which the executed gesture was identified by the system as valid, although not being the one intended by the user, resulted in dynamics and *tempo* changes, altering the particles and audio output. With regard to dynamics, the occurrence of FN (every time the system failed to recognize the executed gesture as binary, ternary, or quaternary) resulted in the particles and audio output behaving only according to the MIDI input. The *tempo* of the execution was not affected by FN, since its value is stored and can only be modified by a valid gesture.

FN happened at least once, for each run segment, regardless of the type of gesture, in 40% of the tests. Considering the total of gestures performed in all tests and disregarding the type of gesture, we found that the incidence rate of FN was 13.33% for Segments 1 and 3, and 16.66% for Segment 2. Observing the number of FN per beat pattern, we conclude that more complex gestures (composed of more movements) cause a higher incidence of FN. We also noted, through the analysis of Segment 1, that the change of beat patterns in the same segment does not imply a higher incidence of FN.

FP occurred at least once, in 10%, 40%, and 20% of the executions of Segments 1, 2 and 3, respectively. Considering the total of gestures performed in all tests and disregarding the type of gesture, the incidence rate of FP was only 1.66%, 6.66%, and 3.33% for Segments 1, 2, and 3, respectively. When observing the number of FP by gesture, the four-beat pattern showed a higher incidence, followed by the two-beat pattern. In addition, when analyzing Segment 2, we conclude that the change of gestures in the same segment does not imply a higher incidence of FP.

TABLE I
SUCCESSFULLY RECOGNIZED GESTURES, FALSE POSITIVES (FP) AND FALSE NEGATIVES (FN) FOR THE RIGHT HAND.

			Segment 1			
	Q	Q	B	T	Q	Q
Run$_1$	✓	✓	✓	✓	✓	✓
Run$_2$	✓	✓	✓	✓	✓	✓
Run$_3$	✓	✓	✓	✓	FP	✓
Run$_4$	✓	✓	✓	✓	✓	✓
Run$_5$	✓	✓	FN	✓	✓	✓
Run$_6$	✓	✓	✓	✓	✓	✓
Run$_7$	FN	✓	✓	✓	✓	✓
Run$_8$	✓	✓	✓	✓	✓	✓
Run$_9$	FN	✓	✓	✓	✓	✓
Run$_{10}$	✓	FN	✓	✓	✓	✓
			Segment 2			
	Q	Q	Q	B	Q	Q
Run$_1$	✓	✓	FN	✓	✓	✓
Run$_2$	✓	✓	✓	✓	FP	✓
Run$_3$	✓	FN	FN	✓	✓	✓
Run$_4$	✓	✓	✓	✓	✓	✓
Run$_5$	✓	✓	✓	✓	✓	✓
Run$_6$	✓	FN	✓	✓	FP	✓
Run$_7$	✓	✓	✓	✓	✓	✓
Run$_8$	✓	FP	✓	✓	✓	✓
Run$_9$	✓	✓	✓	✓	FN	✓
Run$_{10}$	✓	✓	✓	FP	✓	✓
			Segment 3			
	CQ	CQ	CQ	CB	CQ	CQ
Run$_1$	✓	FN	FP	✓	✓	✓
Run$_2$	✓	✓	✓	✓	✓	✓
Run$_3$	✓	✓	✓	✓	✓	✓
Run$_4$	✓	✓	✓	✓	FN	✓
Run$_5$	✓	✓	✓	✓	✓	✓
Run$_6$	FN	✓	✓	✓	✓	✓
Run$_7$	✓	✓	✓	✓	✓	✓
Run$_8$	✓	✓	✓	✓	✓	✓
Run$_9$	✓	FP	✓	✓	✓	✓
Run$_{10}$	✓	✓	✓	✓	FN	✓

Both Segments 2 and 3 test the same sequence of gestures, however, they differ in *tempo*, since Segment 2 is quite faster in the composition, and Segment 3 uses the same gestures to conduct compound time signatures, which means that the movements are slower because each one represents multiple beats. Our results seem also to indicate a direct relation between speed and the occurrence of both FP and FN.

For the left hand tests, a sequence of directions was established, according to different cues in the musical piece, following the melody from section to section, as follows: L, B - 1st Violin, R,T - Viola, R, B - Cello, C, B - Clarinet, L, T - 2nd Violin, C, T - Horn. This sequence of directions was executed 10 times, and the occurrence of FN (when no section was selected) and FP (when a section, different from the one intended, was selected), is shown in Table II:

The occurrence of FN seems to happen predominantly when changing directions in both x and y axes, as it only was detected when changing from LB to RT; and of FP seems to

TABLE II
SUCCESSFULLY SELECTED SECTIONS, FALSE POSITIVES (FP) AND FALSE NEGATIVES (FN) FOR THE LEFT HAND.

	Directions					
	LB	RT	RB	CB	LT	CT
Run$_1$	✓	✓	✓	✓	✓	FP
Run$_2$	✓	✓	✓	✓	✓	✓
Run$_3$	✓	FN	✓	✓	✓	✓
Run$_4$	✓	✓	✓	FP	✓	✓
Run$_5$	✓	FN	✓	✓	✓	✓
Run$_6$	✓	✓	✓	FP	✓	✓
Run$_7$	✓	✓	✓	✓	✓	✓
Run$_8$	✓	✓	✓	✓	✓	✓
Run$_9$	FP	✓	✓	✓	✓	✓
Run$_{10}$	✓	FN	✓	✓	✓	✓

be associated with a proximity in the x-axis, since it occurred when changing from RB to CB and also from LT to CT.

To identify deviations between the positions of the regions of expected gestures and the positions of gestures effectively performed by the user (Figure 13), we considered one single run of each three expected gestures (stored in the array structure), along with the average of 3 perfect executions (with a 100% of success rate) for each of the three gestures performed by the user.

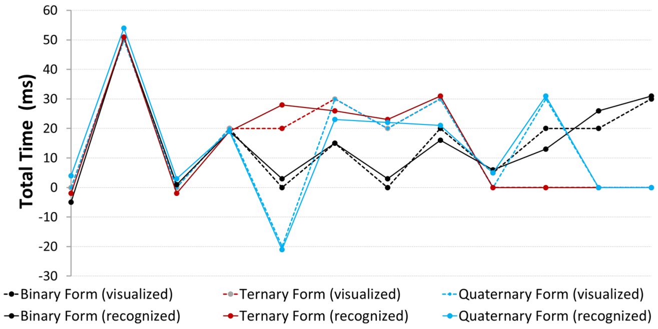

Fig. 13: Diagram of deviations between the positions of expected and recognized gestures.

The existing deviations between the positions of the expected orchestral conducting gestures and the ones actually performed by the user, can be classified according to a threshold L: negligible when $L <= 3$; and small, when $3 <= L <= 6$. In Figure 13, we can observe that the initial and final positions of the three gestures actually performed are very similar to the expected positions, with negligible positional deviations ($L <= 3$). As for the intermediate positions of the gestures, a larger variation was perceived, with small positional deviations ($3 <= L <= 6$). In our system, the classification of the positional deviations has no effect on the gesture executed, since those deviations exist in conducting, usually being signs of expressiveness, besides the fact that this was not the object of study of our work.

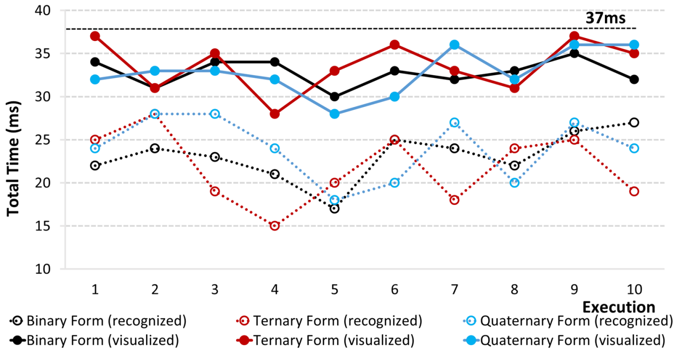

Fig. 14: Latency between the recognized gestures using the *Leap Motion* and the visualized gestures displayed in the computer screen.

As previously mentioned, the identification of gestures (binary, ternary, and quaternary) is made from the analysis of an array data structure containing all the regions through which the hand passed. In order to observe the latency between the time spent to identify each of the three gestures and the visual display on the computer screen (Figure 3), we tested 12, 14 and 15 movements with binary gestures, ternary and quaternary, respectively, in such a way as to get 10 execution runs with a 100% of hit rate. While performing these gestures, we stored the time spent to identify them (from the vector containing the hand positions) and the time spent to render the particles on the computer screen, as shown in Figure 14. In the worst scenario, the three gestures consumed a total time (including recognition and rendering) of $37ms$, a fairly small value for a code in Java, enabling the perception of interactive and synchronous rates, with the gestures captured with the *Leap Motion Controller*.

VI. CONCLUSIONS AND FUTURE WORK

We believe that visualizing a virtual orchestra, in the form of a set of particles emitters, can assist conductors in calibrating their movements, particularly when controlling *tempo*, multiple instruments and dynamics simultaneously.

We explored different forms of visual feedback in a musical performance using three conducting gestures (binary, ternary and quaternary) and one gesture for cutoff, generating a great sense of immersion through gestural control using the *Leap Motion*, in a simple effective way, and at an affordable cost. Furthermore, the possibility of selecting one section of the orchestra offered more freedom to explore new dynamics within the musical execution of a piece.

In the executed tests, we demonstrated the possibility of visually simulating different instruments that have the same note on the score, however, play different sounds due to their different clefs and, thus, emit particles with different colors. Furthermore, our system simulated instruments emitting particles of the same color and sound, even though having the same clef and different notes in the score, because of their transposing musical characteristic. The relation between the amplitude of the conductor's movements with the right hand and the resulting dynamics of the musical performance was also detected, both in the orchestra as a whole and with regard to specific sections selected by the conductor's left hand.

We also conducted performance analysis focusing on the occurrence of FN, FP, positional deviations and latency. In general, the success rate of our implemented solution in gesture recognition was high. Nevertheless, we aim to improve our results by testing new models of trajectories to represent different conducting patterns. Gestures composed of more complex movements generated a higher incidence of FN. The quaternary gesture showed the highest incidence of FP, followed by the binary gesture. Further, the change of gestures in the same segment does not necessarily implied a higher incidence of FN or FP.

With regard to positional deviations, they were almost negligible in the initial and final positions of the gestures, being slightly more remarkable, although still minimum, in the intermediate positions of the curves that represent their paths. The classification of positional deviation has no effect on the performed gesture, since in conducting deviations are signs of expressiveness, which is not the subject of this work. The musical expressiveness concerns the relationship between the articulation of the conductor's gesture and its effect on how a note is played. We designed all the conductor's gestures as *non-espressivo* or without articulation, following the beat pattern without any variations.

Finally, we hope that this prototype can evolve into a support tool for conducting, through which different performances of the same musical piece, or different pieces, might illustrate refinements of conducting movements and their correlations with the behavior of musical instruments. This can happen through the exploration of new forms of gestural control, providing a more natural connection between the gestural input parameters and the resulting visual representation. An implementation of our system using C++ is also planned to verify whether it can offer any advantages and/or more competitive results. Among other future possibilities, new functionalities can be added to our system. For example, we can map wrist movements of the right hand joint (such as articulations), evaluate the best way to visually represent the family of percussion instruments since most of them do not have an indication of pitch, test a new *Leap Motion* left hand calibration accuracy, enabling musical pieces written for larger ensembles with more instruments, etc.

VII. Acknowledgements

We would like to thank Lucile Horn for donating her time and expertise as a professional consultant conductor, and for participating in the user's study evaluating and making insightful suggestions that will help to improve our system. We are also grateful to the referees for providing constructive comments and suggestions to improve the manuscript.

References

[1] S. Abe, *Support Vector Machines for Pattern Classification*. Springer, 2010.

[2] D. Bachmann, F. Weichert, and G. Rinkenauer, "Evaluation of the Leap Motion Controller as a New Contact-Free Pointing Device", *Sensors*, vol. 15, pp. 214–233, 2015.

[3] L. Baraldi, F. Paci, G. Serra, L. Benini, and R. Cucchiara, "Gesture Recognition in Ego-Centric Videos using Dense Trajectories and Hand Segmentation", in *Proceedings of the 2014 IEEE Conf. on Computer Vision and Pattern Recognition Workshops*. IEEE, 2014, pp. 702–707.

[4] G. Bevin, "GECO: Multidimentional MIDI Expression Through Hand Gestures", Available at http://uwyn.com/geco/, last visited March, 2015.

[5] E. Catmull and R. Rom, "A Class of Local Interpolating Splines", *Computer Aided Geometric Design*, pp. 317–326, 1974.

[6] L. M. Controller, Available at https://www.leapmotion.com/product, last visited May, 2014.

[7] ——, Available at https://developer.leapmotion.com/-documentation/csharp/index.html, last visited March, 2015.

[8] R. Demaree Jr. and D. V. Moses, *The Complete Conductor: A Comprehensive Resource for the Professional Conductor of the Twenty-First Century*. Prentice Hall Press, 1994.

[9] J. H. Fonteles, E. S. Silva, and M. A. F. Rodrigues, "Performance Visual Interativa de Partículas Regidas pelo Leap Motion para um Arranjo de Orquestras", in *Proceedings of the XVII Symposium on Virtual Reality and Augmented Reality*, 2015.

[10] G. Glonek and M. Pietruszka, "Natural User Interfaces (NUI): Review", *Journal of the American College of Surgeons*, pp. 27–46, 2012.

[11] L. Hantrakul and K. Kaczmarek, "Implementations of the Leap Motion Device in Sound Synthesis and Interactive Live Performance", in *Proceedings of the 2014 International Workshop on Movement and Computing*, 2014, pp. 142–145.

[12] ——, "Implementations of the Leap Motion Device in Sound Synthesis, Effects Modulation and Assistive Performance Tools", in *Proceedings of the 2014 ICMC*, 2014, pp. 648–653.

[13] H. Hodson, "Leap Motion Hacks Show Potential of new Gesture Tech", *New Scientist*, vol. 218, 2013.

[14] S. Kratz and R. Ballagas, "Unravelling Seams: Improving Mobile Gesture Recognition with Visual Feedback Techniques", in *Proceedings of the 2009 Conference on Human Factors in Computing Systems*. ACM, 2009, pp. 937–940.

[15] J. LaViola Jr., "3D Gestural Interaction: The State of the Field", *ISRN Artificial Intelligence*, vol. 2013, pp. 1–18, 2013.

[16] G. Luck and J. Sloboda, "Exploring the Spatio-Temporal Properties of Simple Conducting Gestures using a Synchronization Task", *Music Perception: An Interdisciplinary Journal*, vol. 25, pp. 225–239, 2008.

[17] M. Mainsbridge, "Non-tactile Gestural Control in Musical Performance", in *Proceedings of the 2014 International Workshop on Movement and Computing*, 2014.

[18] M. R. Malgireddy, J. J. Corso, S. Setlur, V. Govindaraju, and D. Mandalapu, "A Framework for Hand Gesture Recognition and Spotting Using Sub-Gesture Modeling", in *Proceedings of the* 20^{th} *International Conference on Pattern Recognition*. IEEE, 2010, pp. 3780–3783.

[19] G. Meier, *The Score, the Orchestra, and the Conductor*. Oxford University Press, 2009.

[20] J. Oh, T. Kim, and H. Hong, "Using Binary Decision Tree and Multiclass SVM for Human Gesture Recognition", in *Proceedings of the 2013 International Conference on Information Science and Applications (ICISA)*. IEEE, 2013, pp. 1–4.

[21] K. O'Hara, G. Gonzalez, A. Sellen, G. Penney, A. Varnavas, H. Mentis, A. Criminisi, R. Corish, M. Rouncefield, N. Dastur *et al.*, "Touchless Interaction in Surgery", *Communications of the ACM*, pp. 70–77, 2014.

[22] Z. Ren, J. Yuan, J. Meng, and Z. Zhang, "Robust Part-based Hand Gesture Recognition Using Kinect Sensor", *IEEE Transactions on Multimedia*, vol. 15, no. 5, pp. 1110–1120, 2013.

[23] M. C. B. Seixas, J. Cardoso, and M. Dias, "One Hand or Two Hands? 2D Selection Tasks with the Leap Motion Device", in *Proceedings of the* 8^{th} *International Conference on Advances in Computer-Human Interactions (ACHI)*. IARIA, 2015, pp. 33–38.

[24] E. S. Silva, J. A. O. de Abreu, J. H. P. de Almeida, V. Teichrieb, and G. L. Ramalho, "A Preliminary Evaluation of the Leap Motion Sensor as Controller of New Digital Musical Instruments", in *Proceedings of the* 14^{th} *Brazilian Symposium on Computer Music*, 2013, pp. 59–71.

[25] Y. A. Yusoff, A. H. Basori, and F. Mohamed, "Interactive Hand and Arm Gesture Control for 2D Medical Image and 3D Volumetric Medical Visualization", in *Proceedings of the* 9^{th} *International Conference on Cognitive Science*. Elsevier, 2013, pp. 723–729.