Interactive Music Science Collaborative Activities

Team Teaching for STEAM Education

Deliverable 4.7
Final Version of Gesture and VR multimodal interaction interface

Date:	18/07/2018
Author(s):	Aggelos Gkiokas (ATHENA), Kosmas Kritsis (ATHENA), Nancy Zlatintsi (ATHENA)
Contributor(s):	
Quality Assuror(s):	Petros Stergiopoulos (EA), Robert Piéchaud (IRCAM)
Dissemination level:	PU
Work package	WP4 – Core enabling technologies modules of iMuSciCA
Version:	1.0
Keywords:	gesture interaction, multimodal interaction, leap motion, kinect
Description:	Final version of the demonstrator for gesture and VR multimodal interaction interface.

H2020-ICT-22-2016 Technologies for Learning and Skills
iMuSciCA (Interactive Music Science Collaborative Activities)
Project No. 731861
Project Runtime: January 2017 – June 2019

Executive Summary

In this deliverable we present the final version of the demonstrator for gesture and VR multimodal interaction interface. These tools can be summarized into two different categories according to the sensor being used:

- The **Leap Motion** sensor which allows interaction with the music instrument by using the fingers.
- The **Kinect** sensor which allows with the music instrument by using the arms.

Each virtual music instrument is performed with predefined musical gestures, by using one of these sensors. The two demonstrators can be accessed at:

Kinect Demonstrator:
https://athena.imuscica.eu/demos/kinect/v3/

Leap Motion Demonstrator:
https://athena.imuscica.eu/demos/leap/v3/

Software Dependencies:
Leap Motion SDK:
https://developer.leapmotion.com/sdk/v2

Kinect SDK:
https://www.microsoft.com/en-us/download/details.aspx?id=44561

Kinect Additional Software Needed:
https://athena.imuscica.eu/software/kinect/v2/kinectImuscica.rar

Version Log			
Date	**Version No.**	**Author**	**Change**
04-06-2018	0.1	Aggelos Gkiokas (ATHENA)	Initial content
29-06-2018	0.2	Aggelos Gkiokas, Kosmas Kritsis, Nancy Zlatintsi (ATHENA)	Input on different sections
11-07-2018	0.3	Aggelos Gkiokas (ATHENA)	Finalized content and sent for internal review
18-07-2018	1.0	Vassilis Katsouros (ATHENA)	Revision addressing reviewers' comments
18-07-2018	1.0	Vassilis Katsouros (ATHENA)	Submission to the EU

Disclaimer

This publication has been produced with the assistance of the European Union. The content of this publication is the sole responsibility of iMuSciCA consortium and can in no way be taken to reflect the views of the European Union.
iMuSciCA is an H2020 project funded by the European Union.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

Abbreviation	Description
FOV	Field-of-View
HTML	Hypertext Markup Language
OS	Operating System
SDK	Software Development Kit
LSTM	Long Short-Term Memory
RNN	Recurrent Neural Network
ATHENA	ATHENA RESEARCH AND INNOVATION CENTER IN INFORMATION COMMUNICATION & KNOWLEDGE TECHNOLOGIES
UCLL	UC LIMBURG
EA	ELLINOGERMANIKI AGOGI SCHOLI PANAGEA SAVVA AE
IRCAM	INSTITUT DE RECHERCHE ET DE COORDINATION ACOUSTIQUE MUSIQUE
LEOPOLY	3D FOR ALL SZAMITASTECHNIKAI FEJLESZTO KFT
CABRI	Cabrilog SAS
WIRIS	MATHS FOR MORE SL
UNIFRI	UNIVERSITE DE FRIBOURG

1. Introduction

In this deliverable the final version of the gesture interaction with the Virtual Instruments will be described. The aim of gesture interaction core technology is to provide the students (especially those that are not musically educated) with the ability to perform various Virtual Instruments. To do so, two types of sensors are used in the first version of this core technology which are described: The Leap Motion[1] sensor and the Kinect[2] sensor. More details for these sensors can be found in *Deliverable 4.1 - First Version of Gesture and VR multimodal interaction interface*. After extensive experimentation, the free interaction proved to be non-intuitive for students, in contrast to the gesture based interaction (see Deliverable *D5.6 - First Version of Usability validation of iMuSciCA toolkits*). Consequently, in contrast to the interactions described in Deliverable 4.1, the free interaction mode is omitted, and only gesture based interaction are considered. The *Gesture-based interaction* enables the students to play the virtual instrument by performing specific predefined gestures simulating and resembling real instrument playing movements of the arms.

The current deliverable is a demonstrator of the Core Enabling Technology of Gesture Interaction and does not include any connection to any sound creation module.

2. Installation and technical requirements

Both sensors require to go through an installation process for their appropriate SDKs. For the Kinect Sensor, an additional software is needed.

2.1. Installation of the Leap Motion Sensor

For the installation of the Leap Motion Sensor please refer to Section 2.1 of Deliverable 4.1 - First Version of Gesture and VR multimodal interaction interface.

2.2. Installation of the Kinect Sensor

For the installation of the Kinect Sensor please refer to Section 2.2 of Deliverable 4.1 - First Version of Gesture and VR multimodal interaction interface. The difference is that the user has to download an executable program that tracks the skeleton data and additionally recognizes gestures. The program can be downloaded at[3]. Unrar into a folder and run kinectImuscica.exe.

[1] Leap Motion homepage: https://www.leapmotion.com/
[2] Kinect V2 sensor: https://www.xbox.com/en-US/xbox-one/accessories/kinect
[3] https://athena.imuscica.eu/performance/kinect/v2.rar

3. Description of demonstrator and user manual

The Gesture and Virtual Reality (G-VR) tools for music interaction allow students to use their hands (using the Leap Motion sensor) or their upper body and arms (using the Kinect sensor) to perform with a virtual instrument. In contrast to the interactions described in Deliverable 4.1, the free interaction mode is omitted, and only gesture based interaction are considered. For each type of virtual instrument, there is a unique interaction type that uses the appropriate sensor (i.e. Leap Motion or Kinect). With VR Multimodal Interaction Interface the students are “embedded” in a virtual world, where they can see avatars of their hands or bodies interacting directly with the virtual instruments, such as virtually plucking the strings and hitting the drum membrane by using the Leap Motion sensor, or playing the guitar with the Kinect sensor, by selecting predefined chords with the left hand on the fretboard of the virtual instrument and hitting the strings with the right hand.

3.1. Gesture Based Interaction with the Leap Motion

In order to provide an intuitive way for interacting with the virtual instruments, we have developed a gesture recognition system for detecting specific musical gestures. The gesture detection is implemented with two independent approaches, namely, a heuristic-based and a Machine Learning approach based on Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN) architectures, which are described as follows. A demonstrator can be found at URL.

3.1.1. Gesture Recognition

3.1.1.1. Gesture Recognition using heuristics

Currently, the core of our gesture recognition system uses a heuristic method for detecting musical gestures in real-time. The targeted hand gesture classes consists of 8 finger pluckings (considering all fingers, excluding thumbs) and 2 hand tappings. As for the finger pluckings, the system considers two line segments for each finger; the first line corresponds to the metacarpal bone of the finger, while the second line represents an imaginary line that starts from the fingertip, and ends at the joint connecting the proximal phalanx with the metacarpal bone (see Figure 3.1). When the user bends his finger with a downward direction to the palm, the angle between the two lines is computed and compared with a threshold variable. The finger plucking gesture is recognized when the computed angle is smaller than the specified threshold.
Regarding the hand tapping gesture, the heuristic approach takes under account only the fingertip of the middle finger, represented by a 3D point in the virtual 3D space. When the user moves his palm with a downward direction, the system compares the height position of the 3D point with a threshold variable. Finally, the palm tapping is recognized when the height of the 3D point becomes smaller than the threshold.

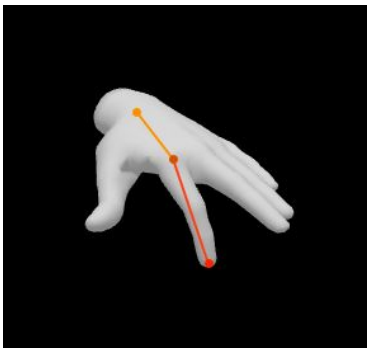


Figure 3.1.1. An example of the considered lines for detecting heuristically the plucking gesture of the left hand's index finger.

3.1.1.2. Gesture Recognition using LSTM Networks

Our gesture recognition approach using LSTM RNN architecture for real-time musical gesture detection, considers as input the raw hand tracking data from the Leap Motion sensor and it is capable to continuously classify sliding windows into the targeted gesture classes, which are related to the responsive interactions that are used during a musical performance with the considered virtual 3D musical instrument. A detailed description of the proposed method can be found in [1]. The targeted gestures, which are depicted in Figure 3.1.2, consists of 7 distinct classes performed by the right hand, including 5 finger pluckings, the palm tap and the finger tap gestures. Moreover, we consider an eighth "unknown" class, that includes any other hand movement. In terms of responsiveness it is assumed that the gestures can be recognized within a small time interval.

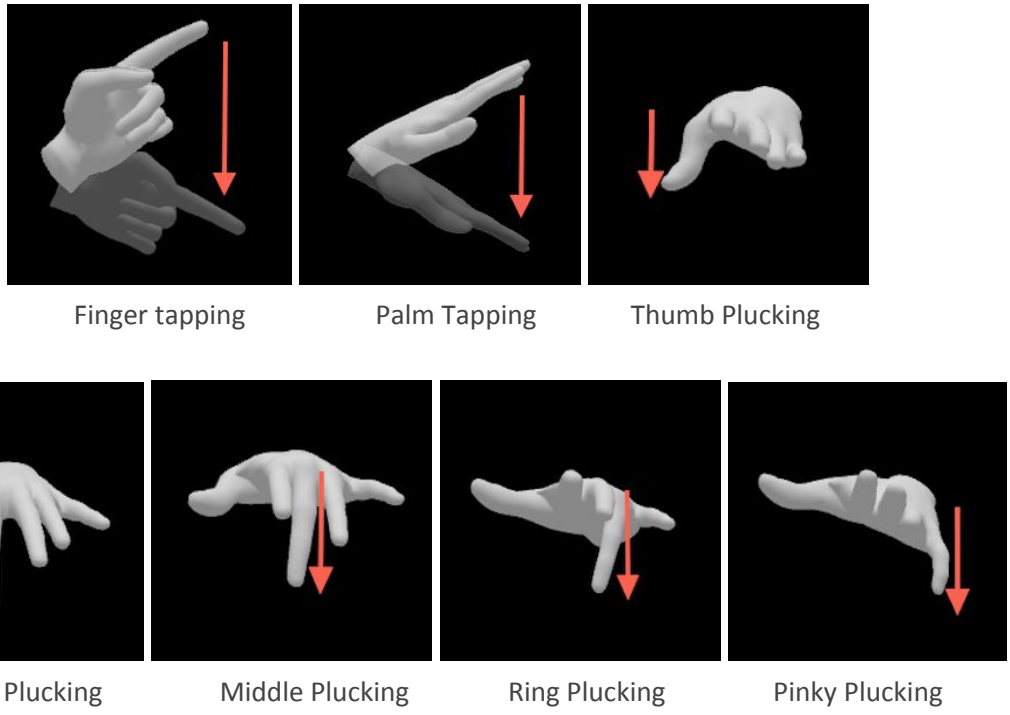


Figure 3.1.2. Illustration of the considered instrumental gesture classes corresponding to the right hand. The temporal evolution of the finger's motion trajectories follows the direction of the arrow.

The system architecture is illustrated in Figure 3.1.3 and uses a Long Short-Term Memory (LSTM) network on top of a feature embedding layer of the raw data sequences as input, which is employed for mapping the input sequence to a vector of fixed dimensionality, that is later passed to a dense layer (fully connected) in order to classify the input data window among the targeted gesture classes. The training of the proposed architecture has been carried out on a dataset collected by 11 participants.

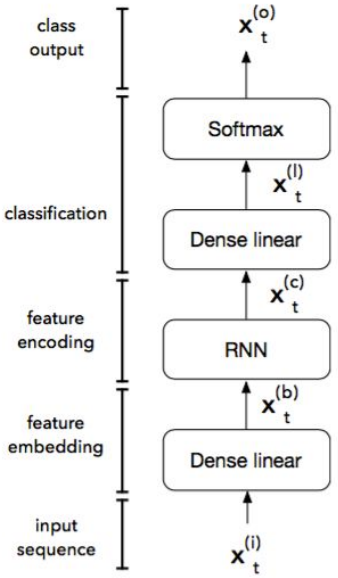


Figure 3.1.3. Architecture of the employed neural network

3.1.1. Interacting with Virtual Music Instruments

Regardless of the backend infrastructure of the gesture recognition system, we deliver three distinct styles for interacting with the different virtual music instruments, including the membrane, xylophone and 2-string instruments, which are presented as follows.

3.1.1.1. Interaction with 2-string Instrument

The 2-string instrument is able to provide up to four different sounding parts by dividing each string with a bridge. Consequently, the user is able to activate a string by performing the plucking gesture. Then, the interaction engine calculates the distance of the distal phalanx of the plucking finger from each string part and activates the closest one.

3.1.1.2. Interaction with membrane Instruments

Even though there are two type of membrane instruments (circular and square membrane), the interaction with these instruments is similar. In order to hit the drum's membrane, the user needs to perform the palm tapping gesture, within the bounds specified by the position and shape of the membrane surface. Then, the interaction engine calculates the impact duration and its coordinates with respect to the membrane surface in order to simulate the mallet up and down movements.

3.1.1.3. Interaction with the xylophone

The interaction engine in the case of the xylophone instrument, allows the user to access up to 8 xylophone bars simultaneously. The eight fingers of both hands (excluding thumbs) are mapped to different bars, starting from the pinky finger of the left hand and ending to the pinky finger of the right hand with an increasing pitch order (from lower to higher bar pitches). Each mapping is activated when the corresponding finger performs the plucking gesture, irrespectively to the 3D position of the virtual instrument.

3.2. Gesture Based Interaction with the Kinect

The demonstrators of the interaction with the Kinect sensor can be found at: [4]Before testing the demonstrator please make sure that the Kinect sensor is plugged to the computer and properly installed. The user has to run the executable and to stand in front of the Kinect camera and position him/herself as if he/she is holding the instrument that is overlaid in the 3D world. Next, the user can test the demonstrator by performing the predefined gestures, which are described in Sec. 3.2.2. A detailed description of the methods used can be found in [2], [3].

3.2.1. Gesture Recognition

For the gesture recognition the skeletons provided by the Kinect are used. Specifically, the application uses all 25 joint positions that are provided by the Kinect v2, to draw a full body 3D virtual avatar that is used in order to improve the interaction and the user interface. In addition, specific joints (such as the position of the hands) are used for recognition of specific gestures that, depending on the selected mode of interaction, generate music.

System Architecture

The Kinect demonstrator consists of two concrete modules: The server, which handles receiving data from the Kinect v2 sensor and sending them in an appropriate JSON format via a Websocket, and the client, which runs in the user's browser and handles the visualization, the sound synthesis and the User Interface as a whole (e.g., user settings). The server part is implemented in the C# language and leverages the Kinect v2 API, in order to receive skeletal information from the Kinect at 30fps. The skeletal information is then converted to an intermediate JSON format, appropriate for transfer via a Websocket. Since we only transfer skeletal information and not the other Kinect streams, the created JSON has a negligible memory footprint and there is no bottleneck regarding the bandwidth of the user's connection. Consequently, there is no delay in transferring the data to the client, even in the case of multiple users playing simultaneously. On the client side, the 3D world that depicts the user and the instruments is built using the three.js library2 which provides, among others, a WebGL renderer for lightweight 3D drawing. The client application, upon receiving the skeletal information, passes the information to both the sound synthesis engine and the visualization engine, which process the movements of the skeleton and output their result to the user's speakers and browser, respectively. The architecture of the Kinect web-based gesture interaction with the virtual instruments is shown in Fig. 3.2.1.

[4] https://athena.imuscia.eu/kinect/demo.html

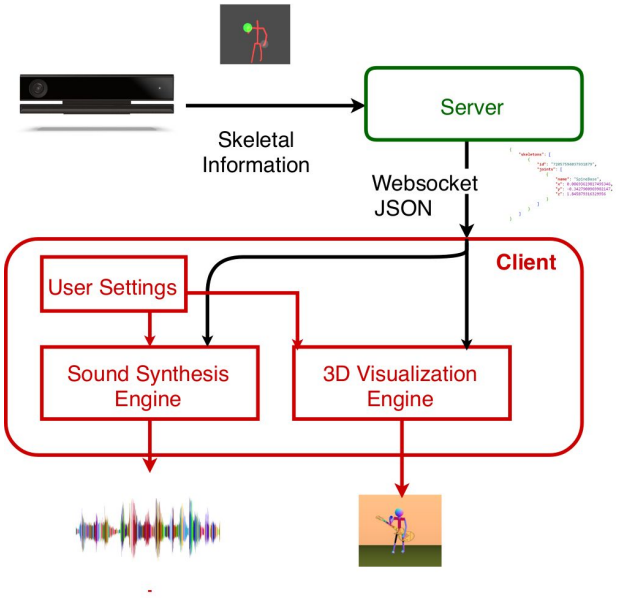


Figure 3.2.1: Architecture of the Kinect web-based gesture interaction with virtual instruments.

3D Visualization Engine

The visualization engine maps the world coordinates (x,y,z) that are received for each skeletal joint directly to the joints of the 3D world avatar in the same metric units. Upon receiving new skeletal information, the engine checks if the skeleton is already in the 3D world and, depending on that, updates the user's joints or creates a new skeleton. If a user stops getting tracked, then the engine removes the skeleton from the world. Depending on the mode that is selected, the 3D visualization engine also renders semi-transparent virtual instruments, and overlaid colored bars with letters that denote the notes that are played.

3.2.2. Interacting with Virtual Instruments

The gesture-based interaction with the virtual instruments supports three different modes. Specifically, in the first two modes, our application enables the users to play the virtual instruments by performing specific "motion templates" (thus predefined gestures) that have some referential similarity to the gestures that a musician does when performing the specific instruments.

The virtual instruments accessible from these modes include: i) the air-guitar, from the first mode and ii) the upright bass, from the second mode (played using a virtual bow). In the third mode each hand is assigned with one of the two previously named instruments with the vertical and horizontal displacement of the hands controlling the pitch and volume respectively. In addition to that, the application enables collaborative playing, hence two, or more, users can collaborate and create music together, either by using the same virtual instrument (from the ones mentioned above) or different ones. Finally, in order to facilitate and guide the interaction of the user, as well as improve the visualization of music generation, each mode provides specific visual aids, apart from the virtual instruments: colored bars accompanied with letters, which show to the user which note is going to be generated next if he performs a sound activation gesture. According to the mode, the associated hands of the user are colored likewise. The different modes and the gesture templates depend on the selected instrument type and are described in more detail next. For more information about the gesture interaction with the Virtual Instruments, we refer the reader to [2].

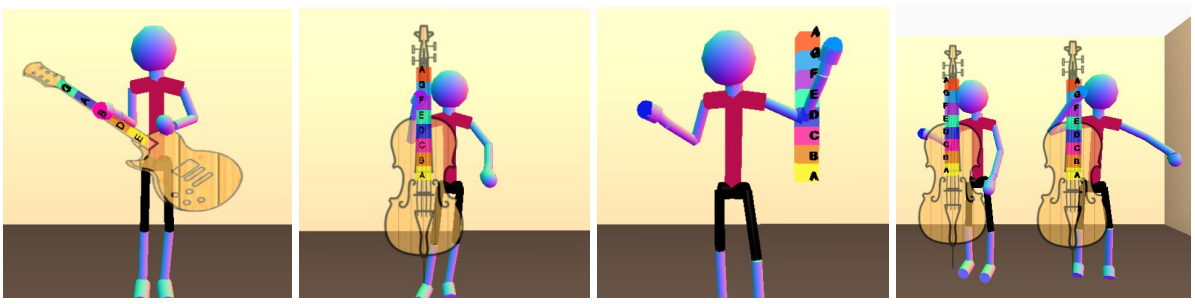
3.2.2.1 Air Guitar Interaction

The user, by selecting the air guitar mode, will be able to perform gestures similar to ones that a guitar-player does.
Gesture 1: In order to enable and activate the "sound" the user brings the right hand around the waist height and moves it vertically (downwards or upwards), simulating this way the moving hand of a guitar player, using as for instance a plectrum. As long as the right hand is performing the specific movement/gesture, a pitch is simulated.
Gesture 2: In order to be able to change the "sounding" pitch of the string, the user has to move the left hand diagonally from the height of the head to below the waist, as if she/he stops the string on the fingerboard determining this way the pitch of the fingered note. This particular gesture is enabled only when Gesture 1 is also active.

For the air guitar mode, two different mappings are predefined, hence the different positions of the left hand are mapped to: i) a pentatonic scale including the notes: G4, A4, B4, D4, and E4 or b) to predefined chords, which are D4, F4, G4, G#4, which when played in the correct order, (provided to the user) can simulate the sound of a well-known guitar riff.

Figure 3.2.2 (left) shows a visualization of the gestures that has to be performed, in order to trigger the various notes. In addition, in this mode, a semi-transparent guitar is rendered on the 3D world that follows the user. Moreover, along the guitar fingerboard, a colored bar is overlaid, that consists of different colors and letters that show which note/chord is played at each different position. We also note that the hand of the user that moves along the fingerboard is also colored correspondingly, in order to visualize to the user the note that will be played if he performs Gesture 1 (sound activation). A snapshot of the 3D world visualization that shows this mode can be seen in Fig. 3.2.3a.

Figure 3.2.2: Visualization of the air-guitar (left) and the upright bass (right) gestures for triggering sounding events of various notes. (Left figure) Right hand, showing the vertical movement, that triggers the sound event and Left hand, showing the diagonal movement across the virtual air guitar fretboard, that triggers different pitches. (Right figure) Right hand, showing the horizontal movement, that triggers the sound event and Left hand, showing the vertical movement, that triggers different pitches.



(a) Air guitar (b) Upright bass (c) Two instrument (d) Multiplayer

Figure 3.2.3: The available modes of the Kinect gesture-based interaction and their 3D world visualizations.

3.2.2.2. Upright bass and Bow Interaction

The user by selecting the Upright Bass and the bowing will be able to perform gestures similar to the ones that an upright bass (or a cello) player does.

Gesture 1: In order to enable and activate the "sound", the user should bring the right hand around the waist height and move it horizontally (from right to left and the other way around), simulating this way the bowing movement, as if the bow is drawn horizontally across the string. As long as the right hand is performing the specific continuous movement/gesture, we assume that the bow is in contact with the string and a pitch should be simulated. When no such movement of the right hand is performed, a sound event is not triggered.

Gesture 2: In order to be able to change the pitch of the string of the upright bass the user has to move the left hand vertically (i.e., downwards or upwards) from the head to the waist height, as if it stops the string on the fingerboard, determining this way the pitch of the fingered note. This particular gesture is enabled only when Gesture 1 is active, generating different pitches, in order to trigger the various notes, refer to Fig. 3.2.2 (right figure) for a visualization of the gestures that has to be performed.

The notes that are simulated depending on the position of the user's left hand, from top to bottom of the fingerboard, are the eight notes of a scale; hence A2, B2, C3, D3, E3, F3, G3, and A3. As in the guitar mode, in the Upright Bass mode, the visualization engine renders an Upright Bass that follows the user. Along the fingerboard of the Bass there is also a color bar and letters that denote the played note, in order to facilitate the interaction as before. A snapshot of this mode can be seen in Fig. 3.2.3b.

3.2.2.3. Two Instruments Interaction

In this mode, each of the user's hands is assigned with one of the two instruments (guitar and upright bass). The movements for triggering sound events are the vertical movements of the two hands, which correspond to various pitches, namely, A3, B3, C4, D4, E4, F4, G4 and A4 (from higher to lower heights). The volume of each instrument is altered by considering the horizontal positions of the hands, obtaining a higher volume when the two hands are further apart, while silencing the instruments when the hands are positioned close to the user's spine. In this mode, the user can actually "air-draw" with the hands, listen to consonant and dissonant musical intervals and generally experiment with the virtual music performance in a more engaging and creative way. As in the previous modes, a color bar with note names is shown vertically denoting which notes are played at each different height level. This mode can be seen in Fig. 3.2.3c.

3.2.2.4. Multiplayer Interaction

In order to enable collaboration between various players, the gesture interaction for music performance can allow the collaboration of two or more players. So in this mode, the users can either play virtually the same instrument, i.e., guitar, upright bass (or even the two instruments mode), or choose to play the two different instruments simultaneously, as shown in Fig. 3.2.4. A snapshot that shows two players both playing the upright bass can be seen in Fig. 3.2.3d. The applications allows for more than two players to stand in front of the Kinect, get assigned with an instrument and be instantly able to interact, while the gestures for the performance are the ones previously described, depending on the instrument.

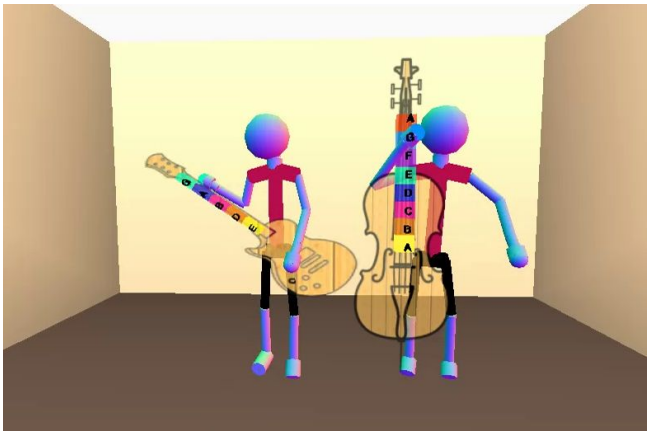


Figure 3.2.4: Snapshot of the collaborative multiplayer interaction with virtual musical instruments.

Another work incorporating two players' interaction, is described in [3], and it regards a performance system for two player, including full-body movements, arm postures and continuous gestures to compose music in real-time. The musical parameters that the performers can influence include the pitch and the volume of the music, the timbre of the sound, as well as the time interval between successive notes. After extensive experimentation between various classifiers, i.e., continuous and discrete Hidden Markov Models (HMM) and a nearest-neighbor classifier (kNN), it was found that the the nearest-neighbor classifier (using geometric representations of the skeleton data as features) outperformed the HMM classifier yielding and accuracy of 98.6%. For the final system and the classification results for continuous gestures and arm postures, the results were also really encouraging and up to 92.11% and 99.33% respectively. For more details we refer the reader to [3].

References

[1] K. Kritsis, A. Gkiokas, M. Kaliakatsos-Papakostas, V. Katsouros and A. Pikrakis, Deployment of LSTMs for Real-Time Hand Gesture Interaction of 3D Virtual Music Instruments with a Leap Motion Sensor", *in Proc. Sound and Music Computing Conference (SMC-2018)*, Limassol, Cyprus, 4-7 July, 2018.

[2] A. Zlatintsi, P.P. Filntisis, C. Garoufis, A. Tsiami, K. Kritsis, M.A. Kaliakatsos-Papakostas, A. Gkiokas, V. Katsouros, and P. Maragos, A Web-based Real-Time Kinect Application for Gestural Interaction

with Virtual Musical Instruments, in Proc. *Audio Mostly Conference (AM'18)*, Wrexham, North Wales, UK, Sep. 2018.

[3] C. Garoufis, A. Zlatintsi and P. Maragos, A Collaborative System for Composing Music via Motion Using a Kinect Sensor and Skeletal Data, in *Proc. Sound and Music Computing Conference (SMC-2018)*, Limassol, Cyprus, 4-7 July, 2018.