Democratizing access to collaborative music making over the network using air instruments

Davide Cocchiara
Department of Information Engineering
and Computer Science, University of Trento
Trento, Italy
davide.cocchiara@unitn.it

Luca Turchet
Department of Information Engineering
and Computer Science, University of Trento
Trento, Italy
luca.turchet@unitn.it

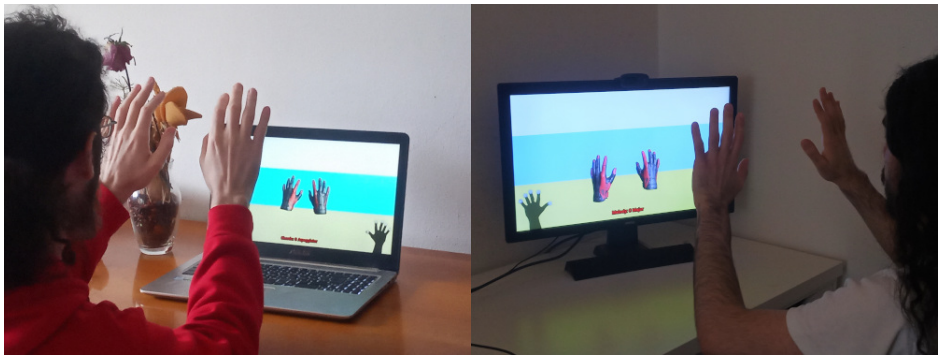


Figure 1: A picture of two connected participants utilizing the developed system together.

ABSTRACT

To date, scarce research has been conducted on the development of tools capable of fostering the democratization of the access to collaborative music making over the network. This paper describes a system based on interconnected air instruments conceived for introducing musically untrained people to collaborative music playing. The system consists of an application controlling synthesizers via real-time finger tracking on input from a consumer-grade camera, which is used in conjunction with a basic networking music performance system communicating control messages. Moving fingers in the air is one of the simplest movements that everybody can afford, thus it was selected as an optimal candidate to build a musical instrument accessible to everybody. A user study was conducted to assess the experience in interacting with the system, involving ten pairs of participants with no musical expertise. Overall, results showed that participants deemed that the system was effective in providing a high user experience, adequate to enable non-musicians to play together at a distance. Moreover, the system was judged as capable of promoting music playing for non-musicians thus fostering easiness of access to music making in a collaborative fashion. A critical reflection on the results is provided, along with a discussion of the study limitations and possible future works.

AM '22, September 6–9, 2022, St. Pölten, Austria

ACM ISBN 978-1-4503-9701-8/22/09…$15.00
https://doi.org/10.1145/3561212.3561227

CCS CONCEPTS

• **Applied computing** → **Sound and music computing**; • **Human-centered computing** → *Web-based interaction*; Sound-based input / output.

KEYWORDS

Networked music performance systems, digital musical instruments, computer vision, Internet of Musical Things

ACM Reference Format:
Davide Cocchiara and Luca Turchet. 2022. Democratizing access to collaborative music making over the network using air instruments. In *AudioMostly 2022 (AM '22), September 6–9, 2022, St. Pölten, Austria.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3561212.3561227

1 INTRODUCTION

Throughout the years, the New Interfaces for Musical Expression community has developed numerous technologies for generating and manipulating music through gestures, exploring new territories not possible with conventional acoustic or electric instruments [5, 16]. Different wearable devices have been utilized to create digital musical instruments relying on the tracking of the user gestures, for instance based on the tracking of muscle activity via biometric sensors [7, 24] or movement quantities via inertial sensors [18, 19]. Non-wearable approaches include the use of motion-capture systems [23], proximity sensors [12], ultrasound waves [14], millimeter waves [2], and cameras via computer vision methods [26].

The use of techniques not relying on wearable devices or tangible interfaces has led to the creation of the so-called "air instruments" [11], where users produce and control sound via gestures in the air. For instance, research in this space has focused on the reproduction of various conventional musical instruments, such as the guitar

[17, 20, 24], drums [26], violin [8], piano [14]. Machine learning (ML) techniques have been widely utilized to build such class of musical instruments, and for this purpose developers have created and used a wide variety of frameworks. Noticeable examples include the Wekinator [9], Gesture Follower [3], Gesture Recognition Toolkit [10] and Gesture Variation Follower [6]. In particular, a significant body of research has focused on the issue of tracking, in real-time, fingers moving in the air (see e.g., [1, 13, 21, 25]).

Playing "in the air" has the potential to be an accessible approach to music making for music beginners or people with no musical expertise [11]. Air instruments can be useful tools to introduce such category of users to music playing, as they can offer a type of control that is naturally bound on the dexterity of users. Also they can be more easily designed to avoid requiring a precision level typically necessary when playing conventional musical instruments, which are designed to support virtuosity. Moreover, since music is a social activity, air instruments can be used to make music together at a distance if coupled with a networked music performance system [22]. For these reasons, this category of digital musical instruments may be a potentially successful candidate for democratizing collaborative music playing over the network. Nevertheless, to the best of authors' knowledge the challenge of using multiple air instruments in conjunction with a networked music performance system to enable geographically displaced users to play collaboratively over the network has not been faced yet.

In this paper we present a system based on interconnected air instruments conceived for introducing to music playing beginners and musically untrained people interested in approaching collaborative music playing. The system consists of an application controlling synthesizers via real-time finger tracking on input from a consumer-grade camera, which is used in conjunction with a basic networking music performance system communicating control messages. Moving fingers in the air is one of the simplest movements that everybody can afford, thus it was selected as an optimal candidate to build a musical instrument accessible to everybody. Therefore, with this study we aimed to provide a tool supporting the democratization of the access to collaborative music making, thus far a scarcely addressed topic in Internet of Musical Things research [27]. Notably, in designing our system we took into account the results reported in [15], where authors found that increasing too much the richness of sound control of a digital musical instrument may result in an instrument less enjoyable to play by non-musicians.

To determine how well our system is capable of supporting people with no musical expertise in playing together, we conducted a user study with twenty users, divided in ten pairs. We addressed two main research questions:

RQ1: Does the developed system provide a high user experience, adequate to enable non-musicians to play together at a distance?

RQ2: To what extent the developed system is capable of promoting music playing for non-musicians?

The developed application[1], the source code[2] as well as a video[3] showing its usage are freely accessible online.

[1]https://github.com/Davide-Cocchiara/MusicalHandsMediaPipe/releases/download/release/MediaPipeInstrumentPublic1.0.zip
[2]https://github.com/Davide-Cocchiara/MusicalHandsMediaPipe
[3]https://youtu.be/AEz-8pCOhGM

2 DESIGN

The first design goal of the system is to let untrained users easily and intuitively play music together through the movement of their fingers. The design focused on the creation of an interactive system which tracks the user's hand including the finger information in real-time. This air instrument allows to play a note by simply closing a single finger. Due to its intuitive interface, it can be used by people with no prior musical experience to play simple melodies and chords, both as lead and as accompaniment.

The second design goal was to let users produce collaborative music that could be perceived as pleasant, consonant, and intuitive to generate and listen to. For this purpose, we designed interactions between two users (although technically, the system supports more than two users simultaneously playing melody or accompaniment), where one user plays melodies while the other produces an accompaniment. Specifically, melodic lines could be built over the major and pentatonic scale of C and G. The notes of the scales, one for each finger for a total of 10 notes, were placed in ascending order of pitch starting from the little finger of the left hand. The two scales were used as they could be perceived as being different from one other, and somehow intuitive to play. Accordingly, the accompaniment was built as arpeggios on the triads of C and G major, where the ten fingers could span three triads, with the lowest note placed on the little finger of the left hand.

The user playing the melody could pass from a scale to the other by moving at least one of the hands on the vertical plane. With the same gesture, the user playing the accompaniment could switch between the arpeggios on the triads of C and G major. Specifically, as detailed in Figure 2, the area of interaction in front of the camera was divided in two regions, colored in yellow (bottom) and blue (top). For the user playing the melody, the bottom-yellow region was associated to the major scale, the top-blue region to the pentatonic scale. For the user playing the accompaniment, the bottom-yellow region was associated to the arpeggios in C major, the top-blue region to the arpeggios in G major. Changing the position of at least one hand from a region to the other would cause the change in the scale.

Figure 2: A screenshot of the two interaction areas used to change key and scale types.

The key of the scales played by the user playing the melody was automatically changed by the key changes performed by user playing the accompaniment, in order to avoid dissonances and keep harmonic coherence. We used different timbres for melody (a piano-like sound) and accompaniment (a pad sound) in order for the users

to perceive immediately what was their own contribution with respect to the one generated by the other user. Moreover, the users were empowered to easily interchange their roles while playing. This was achieved by letting the user press the TAB key on the keyboard.

The third design goal was to create a fully standalone system easy to install and not requiring other software dependencies, given the fact that musically untrained people are typically not acquainted with musical software, such as digital audio workstations (DAWs) or networked music performance systems. The system was designed to work with conventional hardware, such as a laptop with average computing capabilities and a consumer-grade camera. Also, the system was designed to enable interactions over the network using conventional network capabilities in terms of bandwidth and throughput. All this constitutes an approach affordable by a wide variety of users, thus fostering democratization of collaborative music making which was our end goal.

3 IMPLEMENTATION

3.1 System architecture and components

The architecture of the developed system is illustrated in Figure 3. The input consists of image frames acquired from a consumer-grade camera, while the output consists of audio-visual content: the sounds produced locally and those received from the connected user, and the real-time visualization of the hands of the user (see Figure 2).

The system leverages Unreal Engine 4 (UE4) to render a faithful 3D visualization of the pose of the user's hands and to execute the necessary logic to generate the played sounds across networked devices. For finger tracking we utilized Google's MediaPipe Hands ML framework[4] [28]. We wrote a custom UE4 plugin in C++ to read 2D landmarks from MediaPipe, starting from an open source implementation[5]. In UE4, plugins are collections of code and data that developers can easily enable or disable within the Editor on a per-project basis. Plugins can add runtime functionality, and add or modify built-in engine features.

The MediaPipe's finger tracking algorithm returns 21 2D landmarks for each hand (see Figure 4). However, the tracking algorithm presents some issues. While MediaPipe is capable of giving an estimate of the landmarks' position on the depth axis, we found that such an estimate is not sufficiently accurate for our purpose of controlling sound in real-time, thus we chose to ignore it. The hands' position is only reconstructed along a plane parallel to the camera. MediaPipe also provides an estimate called "Handedness", which tries to determine if a given hand is a left or right hand. We found that this estimate is unreliable and tends to have low temporal coherence, rapidly flickering between left and right hand. Therefore, we retrieved the information about the hands by using their position in space: the leftmost hand relative to the camera is always assumed to be the left hand, and the rightmost the right hand.

A DLL was coded to create a bridge between the UE4 Plugin and MediaPipe. This is the only design choice that binds the system to Windows-only platforms. The system could be ported onto all UE4 supported platforms if a different bridging mechanism was utilized. UE4's supported platforms include Windows, Linux, MacOS, Android, and iOS.

The finger tracking's responsiveness is bound by the minimum frame rate of each component, where the slowest acts as a bottleneck. The slowest point is typically the frame rate of the camera, which is usually between 30-150 FPS. On consumer grade webcams, using the system with abundant lighting is necessary for a smooth experience, since the frame rate is dependent on the camera's ISO sensitivity and aperture. Since UE4 was configured to run on lower-spec hardware, the system is bound by the speed of MediaPipe's ML algorithm.

The logic to interpolate the fingers' 3D positions starting from 2D landmarks was implemented into UE4. We used the UE4's animation system to display the interpolated hands' mesh. Furthermore, we used collision detection to determine if a note had to be triggered. Specifically, we utilized the paradigm of a virtual string that needs to be plucked with the fingers to generate sounds.

Once a note has been determined to be triggered, this information is sent as an Open Sound Control (OSC) message to both a synthesizer and the network. The same process occurs for the information related to the vertical position of the hands, which controls the volume. The OSC message is provided as input to a built-in synthesizer or, alternatively, to an external DAW.

3.1.1 UE4 architecture. Figure 5 details the subcomponents of the UE4 part of the system, and the related data flow. In UE4, objects that are placeable in the 3D world are called "Actors".

Hand Manager. This component requests landmarks from MediaPipe and validates the input. It computes 2D landmarks into 3D transforms by applying the hand reconstruction algorithm. It decides whether the incoming input is valid as well as it decides which hand is left or right. Moreover, it provides 3D transforms to the Player Hands actor.

Player Hands. This component handles the 3D representation in the world of the pose of the player's hands. It contains two skeletal meshes whose pose is computed in real-time by requesting 3D transforms from the Hand Manager actor. UE4's Animation Blueprints are used to compute forward kinematics and bend the bones accordingly. Collision and visibility is enabled or disabled based on information from the Hand Manager actor.

Plucking Instrument. This component implements the paradigm of the finger plucking a virtual string. It receives the information about the collisions from Player Hands and decides when and which note is to be triggered. It also relays this information to the OSC Sender actor.

OSC Sender. This component is responsible for sending OSC messages to the built-in synthesizer, or to an external DAW. It is also responsible for sending the note control information over the network.

Built-in synthesizer. The UE4's built-in synthesizer, allowing to generate piano-like sounds for the melody and pad-like sounds for the accompaniment.

DAW. An (optional) digital audio workstation (e.g., Reaper, Logic) which accepts OSC messages to control parameters of synthesizers plugins. The DAW is alternative to the built-in synthesizer.

[4]https://google.github.io/mediapipe/solutions/hands.html
[5]https://github.com/wongfei/ue4-mediapipe-plugin

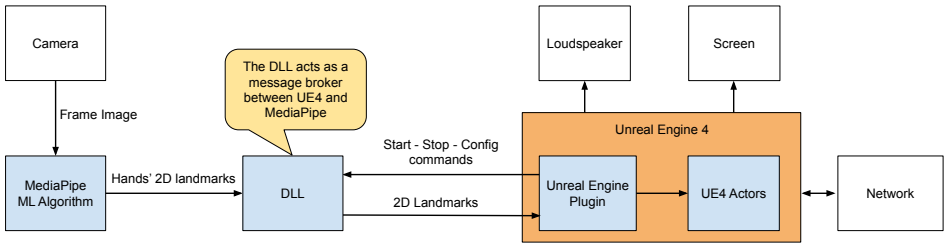
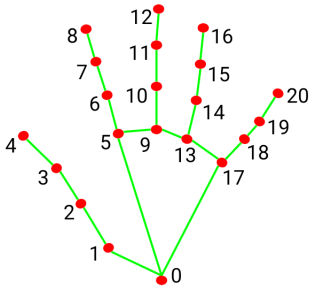


Figure 3: A block diagram detailing the architecture of the system and its components.



0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

Figure 4: The MediaPipe's landmarks.

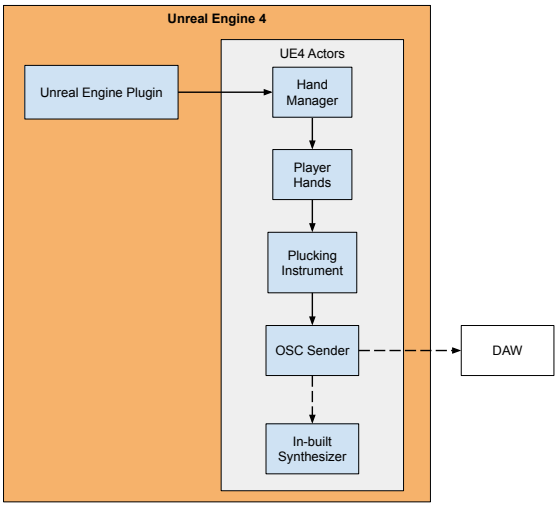


Figure 5: Block diagram of the UE4 component of the system. The dashed lines indicate the two alternative ways to generate sounds, via the built-in synthesizer or an external DAW.

3.2 Hand reconstruction algorithm

A naive approach to reconstruct the hands would be to reconstruct them based on the position of each landmark. Since a conventional webcam does not provide an accurate depth-axis estimation, the hand would appear bigger or smaller based on the distance from the camera. Moreover, since the hands' shapes vary greatly between humans, any 3D model would appear misaligned and different to each user. Therefore, we opted for an approach based on the use of the angle between each landmark.

3D models of left and right hands were prepared using Blender 2.8. Due to the technique employed to reconstruct the hands' position, custom models were rigged appropriately and used in UE4 as skeletal meshes. The models' bones were built to match MediaPipe's landmarks. To calculate the angle between finger bones, the forward vector of each bone also needed to be correctly oriented.

The hands' root positions simply corresponded to their wrist's landmark. To calculate the angles between each finger bone, an iterative process was applied for each finger, starting from the wrist. Knowing that finger bones mostly rotate along two rotational axes, the angle between each landmark was calculated on the pitch and yaw axes. Then, the rotation was applied to the corresponding bone.

3.3 Speed and UE4 Optimization

To allow the project to be run on lower-spec hardware, the UE4's rendering features were reduced to a minimum. Instead of using the default shading model (Shader Model 5.0) we utilized the OpenGL ES 3.1 renderer. This renderer was specifically conceived and developed to be used on mobile devices. It provides less features, but its speed is significantly higher. Such a renderer can also be used on Windows, through DirectX Mobile Emulation. All engine scalability settings were forced to a minimum, where lighting and shadows were disabled. Resolution was constrained to 1200x900.

3.4 Sound synthesis and control

The air instrument is a beginner-friendly instrument designed to be used by placing both hands directly in front of the camera, and with the palms parallel to it. The input is accepted only if hands are placed this way. This is achieved by a filtering process based on the root rotation of both hands, and by checking that both hands

are being currently tracked. When the player closes a finger, a note is played. Visual feedback is provided to the player as shown in Figure 2. Each time a note is triggered, the interface changes the color of the corresponding finger to green.

This instrument uses UE4's collision capsules and boxes (see Figure 6). A box is placed on the palms, and a capsule is placed on each fingertip. The capsules are attached to the finger bones, and move along with them. When a finger's collision capsule intersects with the palm's collision box, a note is played. After the finger's capsule leaves the box, the note can be played again. Collisions are filtered by channel: fingers and palms belong to different channels to avoid useless calls when palms intersect. Collisions are also filtered by hand, where the left hand's fingers cannot be used to trigger the right hand's palm collision box.

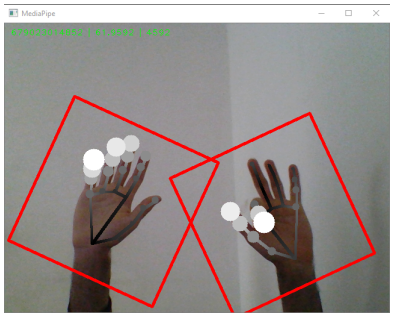


Figure 6: Tracked 3D hand landmarks are represented by dots in different shades, with the brighter ones denoting landmarks closer to the camera. The red boxes represent the detected palms in the image.

Upon recognition of a finger with the box a note is triggered. This is achieved via a mapping between fingers and notes, which depends on the key used (C or G) and the role covered by the user (melody or accompaniment player). When using the UE4's built-in synthesizer, piano-like sounds were generated for the melody and pad-like sounds for the accompaniment. Alternatively, the user can control an external DAW via OSC messages.

3.5 Networking

For the networking part, we relied on a networked music performance system communicating not audio signals [22], but OSC messages. Specifically, the default built-in UE4 client-server architecture was utilized, which relies on UDP protocol with packets having a timestamp[6].

Before being sent to other clients, OSC messages are deconstructed into the minimal amount of information. Specifically, for the notes played, the message is deconstructed into: note, intensity, player producing the note. This gives rise to a message of 3 bytes, plus the payload identifying the message type. Each time a player triggers a note, such a note not only is immediately played on the user device, but also is relayed to the server, which propagates the note to everyone but the player that originally produced that note. The player performing the accompaniment relays a key change with a message of two bytes, plus the payload: new key, invoking player.

[6]https://docs.unrealengine.com/4.27/en-US/InteractiveExperiences/Networking/Overview/

Table 1: F1 score, Precision, Recall resulting from the tests involving 1000 notes.

F1 score	Precision	Recall
0.992	0.987	0.997

4 EVALUATION

4.1 Technical validation

Firstly we assessed the suitability of different hardware in supporting the application and measured the system's speed. Below, we indicated in parentheses the frame rate of UE4 expressed in ms, which represents how fast our system can process a frame. Since the system is also bound by the speed of MediaPipe hands, the reader can refer to Google MediaPipe's documentation for its speed estimates. We conducted tests with the following hardware:

- A modern desktop computer with an AMD 6700 XT video card, which ran with more than 3500 FPS (0.29ms);
- A high performance 2017 laptop with a dedicated NVIDIA GeForce GTX 1050, the ASUS Vivobook Pro 15 N580VD, which ran with more than 350 FPS (2.86ms);
- A low-budget laptop from 2014, the HP 250 G3, could run UE4 smoothly with over 60+ FPS (16.66ms). However, it struggled with keeping up with MediaPipe's machine learning model, making the hand tracking usable but sluggish.

Secondly, we assessed the finger tracking accuracy. For this purpose, we repeatedly played through each note on each finger twice, for a total of a thousand notes. The tests were conducted in optimal lighting conditions, by the authors. Table 1 reports the precision, recall and F1 score. All false positives were notes triggered two times in rapid succession, rather than once. Almost all of them were thumb notes. During the user study, the involved inexperienced participants had a lower accuracy. Simultaneous notes also tend not to work well when fingers occlude one another (e.g., thumb and index of the same hand simultaneously played).

4.2 User experience evaluation

The user study aimed at assessing the usability of the system and participants' experience in interacting with it, with the end goal of answering to our two research questions.

4.2.1 Participants. A total of twenty participants took part to the evaluation (16 males, 4 females, aged between 18 and 64, mean age = 29.2, standard deviation = 12.1). All participants reported to have no prior musical experience in playing an instrument. The experiments were conducted at the home of participants. Fifteen participants were Italian, while five where from other European countries and the US. Participants took on average one hour to complete the experiment. The procedure, approved by the local ethics committee, was in accordance with the ethical standards of the 1964 Declaration of Helsinki.

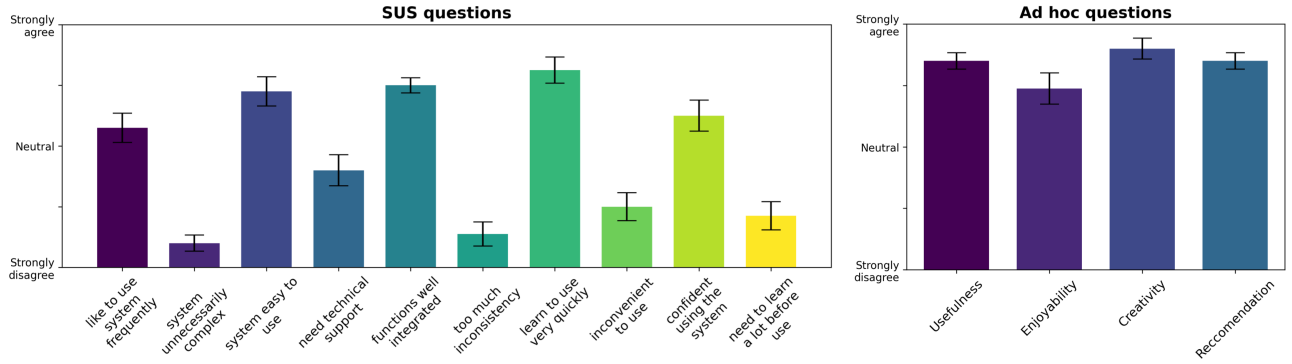


Figure 7: Mean and standard error of the scores of the System Usability Scale topics for the tested system (left), and of the ad hoc questionnaire (right).

4.2.2 Procedure. Participants were divided in ten pairs, which were geographically displaced within a distance ranging between 10 and 200 Km. Each pair was first introduced to the functionalities of the system and then to the tasks to be performed. A familiarization phase followed. This initial part was explained and supervised by a remotely connected experimenter. Participants were instructed to conduct two tasks: in the first task, one participant played the melody-based instrument, spanning both the major and pentatonic scales, while the other participant played the accompaniment; in the second task the roles were inverted. Each task lasted about 15 minutes.

After having used the system, participants were administered a questionnaire comprising different parts. In the first part, we assessed the usability of the system using the well-established System Usability Scale (SUS) questionnaire. The second part consisted of an ad hoc questionnaire composed by the following questions to be evaluated on a 5-point Likert scale (1 = strongly disagree, 5 = strongly agree):

(1) I think that the system is useful to attract non-musicians to the world of played music;
(2) I found the experience of playing with another user enjoyable;
(3) I think that the system is useful to stimulate musical creativity;
(4) I would recommend this system to a non-musician friend.

The third part consisted of the following open-ended questions:

(1) Describe the positive parts of your experience in interacting with the system;
(2) Describe the negative parts of your experience in interacting with the system;
(3) How would you improve on the system?

Finally, participants were given the opportunity to leave an open comment.

4.2.3 Results. **1) SUS questionnaire.** The SUS metric assesses the usability of a system on a scale from 0 to 100. As a point of comparison, an average SUS score of about 68 was obtained from over 500 studies. Our system obtained a mean SUS score of 73.87 (95% confidence interval: [67.55; 80.19]) which is above average. Figure 7 (left) shows the breakdown of the result across the topics of the scale. The results reported in the figure indicates that on average participants found the system easy to use, simple, quick to learn and to use without technical support.

2) Ad hoc questionnaire. Figure 7 (right) shows the mean and standard error of participants' evaluations to the four ad hoc questions. As it is possible to notice, all items were ranked with a high score on average.

3) Open-ended questions. Participants' answers to the open-ended questions were analyzed using an inductive thematic analysis [4]. The analysis was conducted by generating codes, which were further organized into themes that reflected patterns, as described below.

Concept and easiness of use. Six participants commented to have very much appreciated the concept underlying our system, i.e., allowing musically-untrained people to express themselves musically with simple gestures as well as play together (e.g., *"The idea is really fun and stimulating, I'd love to do this with my friends in the future"; "It's a very interesting concept and well implemented"; "I find it a method simple and innovative to allow anyone to access the world of music playing"; "I think this is a wonderful tool for sharing music with people, especially kids, and its fantastic that it requires a pretty minimal level of technology for people to use."*). Eleven participants found the system easy to setup and use, commenting in particular on the naturalness of the interaction hand-sounds (e.g., *"It is easy to get started as there is no set-up or an instrument necessary"; "Being able to instantly play around with hitting notes without any other complexity was neat, and it was fun being able to do so using my hands out in front of me. I also thing it's neat being able to switch between keys and chords."; "I appreciated the fluidity of the sounds with the movements, which creates a kind of coordination hand-ear more simple and immediate with respect to conventional musical instruments"*).

Fun, pleasantness and creativity stimulation. Five participants found that the experience of interacting with the system was fun (e.g., *"I found the fact that I could create music with nothing but fingers and software really fun, it's creatively stimulating"*). Two participants commented that the produced sounds were pleasant, while other two reported that the system is capable of effectively

stimulate their musical creativity (e.g., *"Even if I don't know anything about music, with the movement of my hands I was able to play a melody. It was fun!"; "The system stimulates instinctively the musical creation, it is pleasant in its similarity with a game. Moreover I can play an instrument without the need of purchasing a real musical instrument"*).

Tracking issues. Six participants reported to have sometimes experienced some issues with the tracking system, finding it not optimal. (e.g., *"Initially, I didn't understand how to make the system work because I could not understand how far I had to be from the camera"; "It is not immediate to find the right position of the hand and learn how to do the right movements so the system triggers the notes"; "Sometimes it is not easy to trigger the notes, it is necessary to understand first how much to bend a finger, and if its bending is too much inevitably I will bend also the fingers next to it causing unwanted triggering"*). Four participants suggested to improve the tracking system or notify the user about possible issues in the lighting conditions (e.g., *"I would suggest to add a popup informing the user if the lighting is not correct"*).

Tiring position. Four participants reported that keeping the hands in vertical position could be tiring in the long run (especially for users with wrist issues such as the carpal tunnel), and suggested to improve the system by allowing the tracking with the hands parallel to the ground (e.g., *"It takes some practice to learn the best position for your hands and holding them up can be a bit tiring"; "It would be more comfortable to use the system with the palms facing the ground"*).

Expressivity range extension. Five participants commented that would have loved to have some higher expressivity support, in terms of the generated sounds (e.g., *"I could not change the kind of notes produced"; "I would add the possibility to let the user choose the sounds and the scales types"*).

Missing visual feedback of the other player. Two participants reported some difficulties in playing together with another user, suggesting to improve the system by providing some visual information about the other player rather than being forced in interacting only at auditory level (e.g., *"For the multiplayer aspect I would like to be able to see the hands of the other player; "I also thought that playing with another individual was a little difficult. There was no way to see their intent or communicate, other than listening for the sounds they played, which made it feel like we were both doing separate rather than playing together."; "I think for the two-play system it would be nice to have some sort of representation of what the other player is doing. For example, a second set of hands that appears on the screen, and whose fingertips light up in addition to the sound cues that you already get."*).

Virtual teacher. Six participants provided suggestions on how to improve the user interface of the system with a support for learning it more effectively and also learn music. Firstly, the system could be extended with a virtual assistant teaching how to use it properly (e.g., *"A guide for how you should place your hands and how the camera should be oriented would be helpful"; "It would be useful to have a mini tutorial showing the correct movements to be performed"*). Second, the need for having a visual indication of the mapping between a note and a finger so to learn music was risen (e.g., *"It would be useful to have a real-time visual indication of which note are played"*).

5 DISCUSSION AND CONCLUSIONS

This work aimed at assessing whether it is possible to democratize access to collaborative music making over the network using an easy to setup system based on an air instrument and a basic networked music performance system. The air instrument was based on moving fingers to generate sounds, which is one of the simplest movements that everybody can afford, thus conferring the system with accessibility characteristics. To investigate our research questions RQ1 and RQ2 (see Section 1) we conducted a study with ten pair of users with no musical expertise assessing the experience in interacting with the system and reflections on its potential use. Overall, results showed that participants deemed that the system was effective in providing a high user experience, adequate to enable non-musicians to play together at a distance. Moreover, the system was judged as capable of promoting music playing for non-musicians thus fostering easiness of access to music making in a collaborative fashion.

It was observed that participants took some time to get acquainted with the system despite it was judged intuitive and easy to use and learn. While this is obvious with any system, a complicating factor was the lack of musical expertise. Nevertheless, after the inevitable initial difficulties of understanding how to control the system, participants showed to learn the system quickly and use it meaningfully. It was also observed that participants experienced some initial difficulties in synchronizing, which is ascribable to the lack of musical knowledge. However, after a few minutes of playing together participants could create together meaningful musical interactions.

Most participants commented that the experience was fun and some of them reported even to be enthusiastic. This was due to the fact that the system effectively allowed them to accomplish an activity, the musical one, which was in their desire but that they had not been able to conduct for various reasons (e.g., lack of sufficient musical skills with using conventional musical instruments, costs for musical education). The system was appreciated by all participants for allowing anybody to play music both alone and in collaboration with others. It was deemed by such participants to be effectively capable of promoting music playing for non-musicians. Given the adopted design approach for the sound control, the results reported in this study seem to confirm the findings reported in [15]. Such findings suggested that a digital musical instrument requiring less physical and mental effort from the player can lead to more enjoyable experiences for non-musicians even if it has a more restricted space of possibilities.

The evaluation provided insights also about the negative aspects of our system. The position used for generating the notes was deemed as tiring for long sessions, and participants suggested that a more comfortable position would be that of having the palms facing the ground. However, such interactions style entails a redesign of the system, involving a camera tracking the hands movements from the top or the bottom. This, however, would make the system less accessible as users would not simply need a laptop with a built-in camera, but an external camera to be appropriately placed.

Some participants reported to have experienced some issues with the tracking system. The causes could be manifold. First, the tracking issues could have depended on the non-optimal lighting

conditions of the rooms in which the test were conducted (i.e., at the participants' home). Second, the issues could be related to the non-perfect user's movements. Third the accuracy of the tracking system by default was not 100% accurate according to the technical validation reported in Section 4.1.

Notably, our camera-based approach is less useful in a performance scenario, since it is so dependent on the placement of the camera and on appropriate lighting conditions. The system was not conceived for musicians, but could be complicated to support more expressive features (e.g., volume control, use of different instruments and scale, effects modulations, etc.). An intrinsic limitation of the system lies in the speed of the finger tracking, which is bound to the latencies introduced by the acquisition from the camera (usually between 30-150 FPS) and the processing capabilities of the underlying computer. While none of the users involved in the study reported any comment about movement-to-sound latency perception, it is plausible to expect that musicians would perceive latencies and not tolerate them especially for continuous fast tempo melodies.

We acknowledge that for making even more accessible the app, web technologies could be used in place of a standalone application. It is plausible that the application can be adapted to also run on a web browser using Unreal Engine 4, but for this release we wanted to maximize system stability and hardware accessibility. Indeed, while at present Unreal Engine 4 supports HTML5 experimentally, we wanted to keep the computing resources used and delay to a minimum. We noted that some computers where the app had been run successfully, did not have enough resources to run the application and a modern web browser at the same time.

A study limitation is that we involved a low number of participants (a total of twenty) and all from Western countries. Thus, the generalizability of the reported results remains to be assessed for a wider pool of participants from different countries and musical cultures. Nevertheless, the participants' comments are encouraging and point towards the conclusion that the proposed system has a concrete potential to democratize access to collaborative music making over the network. Furthermore, we believe that the system could find applicability to introduce children to music playing. Another study limitation concerns the fact that the system was tested for a relatively small amount of time. A longer, longitudinal study would unravel more clearly what is the system's actual potential of being used over extended periods of time so that non-musicians can further develop their skills.

In future work we plan to add a video tutorial to instruct users on how to learn the system by themselves. We also plan to include an interactive system providing the users with real-time feedback about how to set up at best the lighting conditions and about indications on how to conduct the movements in an appropriate way. Moreover, we plan to provide users with some simple and familiar reference songs to learn and enjoy playing. Finally, our system is currently built for Windows, but we plan to extend it for other platforms, including the mobile ones.

REFERENCES

[1] M. S. Alam, K. Kwon, and N. Kim. 2021. Implementation of a character recognition system based on finger-joint tracking using a depth camera. *IEEE Transactions on Human-Machine Systems* 51, 3 (2021), 229–241.

[2] F. Bernardo, N. Arner, and P. Batchelor. 2017. O soli mio: exploring millimeter wave radar for musical interaction.. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, Vol. 17. 283–286.

[3] F. Bevilacqua, B. Zamborlin, A. Sypniewski, N. Schnell, F. Guédy, and N. Rasamimanana. 2009. Continuous realtime gesture following and recognition. In *International gesture workshop*. Springer, 73–84.

[4] V. Braun and V. Clarke. 2006. Using thematic analysis in psychology. *Qualitative Research in Psychology* 3, 2 (2006), 77–101.

[5] D. Brown, C. Nash, and T. Mitchell. 2018. Simple mappings, expressive movement: a qualitative investigation into the end-user mapping design of experienced mid-air musicians. *Digital Creativity* 29, 2-3 (2018), 129–148.

[6] B. Caramiaux, N. Montecchio, A. Tanaka, and F. Bevilacqua. 2014. Adaptive gesture recognition with variation estimation for interactive systems. *ACM Transactions on Interactive Intelligent Systems* 4, 4 (2014), 1–34.

[7] C. Erdem, Q. Lan, J. Fuhrer, C. P. Martin, J. Tørresen, and A. R. Jensenius. 2020. Towards Playing in the "Air": Modeling Motion-Sound Energy Relationships in Electric Guitar Performance Using Deep Neural Networks. In *Proceedings of the Sound and Music Computing Conference*. 177–184.

[8] X. Fan and G. Essl. 2013. Air Violin: A Body-centric Style Musical Instrument. In *Proceedings of the International Conference on New Interfaces for Musical Expression*. 122–123.

[9] R. Fiebrink and P. R. Cook. 2010. The Wekinator: a system for real-time, interactive machine learning in music. In *Proceedings of the International Society for Music Information Retrieval Conference*, Vol. 3.

[10] N. Gillian and J. A. Paradiso. 2014. The gesture recognition toolkit. *The Journal of Machine Learning Research* 15, 1 (2014), 3483–3487.

[11] R. I. Godøy, E. Haga, and A. R. Jensenius. 2005. Playing "air instruments": mimicry of sound-producing gestures by novices and experts. In *International Gesture Workshop*. Springer, 256–267.

[12] J. Han and N.E. Gold. 2014. Lessons learned in exploring the Leap Motion™ sensor for gesture-based instrument design. In *Proceedings of the International Conference on New Interfaces for Musical Expression*. Goldsmiths University of London, 371–374.

[13] T. Hongyong and Y. Youling. 2012. Finger tracking and gesture recognition with kinect. In *IEEE International Conference on Computer and Information Technology*. IEEE, 214–218.

[14] I. Hwang, H. Son, and J.R. Kim. 2017. AirPiano: Enhancing music playing experience in virtual reality with mid-air haptic feedback. In *IEEE World Haptics Conference*. IEEE, 213–218.

[15] R. Jack, J. Harrison, F. Morreale, and A. McPherson. 2018. Democratising DMIs: the relationship of expertise and control intimacy. In *Proceedings of the Conference on New Interfaces for Musical Expression*.

[16] A. R. Jensenius. 2017. Sonic Microinteraction in "the Air". In *The Routledge Companion to Embodied Music Interaction*, M. Lesaffre, P.-J. Maes, and M. Leman (Eds.). Routledge, 431–439.

[17] M. Karjalainen, T. Mäki-Patola, A. Kanerva, and A. Huovilainen. 2006. Virtual air guitar. *Journal of the Audio Engineering Society* 54, 10 (2006), 964–980.

[18] D. Keller, C. Gomes, and L. Aliel. 2018. The Handy Metaphor: Bimanual, touchless interaction for the Internet of Musical Things. In *Proceedings of the Eight Workshop on Ubiquitous Music*. 180–188.

[19] T. J. Mitchell, S. Madgwick, and I. Heap. 2012. Musical interaction with hand posture and orientation: A toolbox of gestural control mechanisms. In *Proceedings of the International Conference on New Interfaces for Musical Expression*.

[20] J. Pakarinen, T. Puputti, and V. Välimäki. 2008. Virtual slide guitar. *Computer Music Journal* 32, 3 (2008), 42–54.

[21] K. Park, S. Kim, Y. Yoon, T. Kim, and G. Lee. 2020. DeepFisheye: Near-surface multi-finger tracking technology using fisheye camera. In *Proceedings of the Annual ACM Symposium on User Interface Software and Technology*. 1132–1146.

[22] C. Rottondi, C. Chafe, C. Allocchio, and A. Sarti. 2016. An Overview on Networked Music Performance Technologies. *IEEE Access* 4 (2016), 8823–8843.

[23] S. A. Skogstad, A. R. Jensenius, and K. Nymoen. 2010. Using IR optical marker based motion capture for exploring musical interaction. In *Proceedings of the International Conference on New Interfaces for Musical Expression*. 407–410.

[24] J. E. Tamani, J. C. B. Cruz, J. R. Cruzada, J. Valenzuela, K. G. Chan, and J. A. Deja. 2018. Building guitar strum models for an interactive air guitar prototype. In *Proceedings of the 4th International Conference on Human-Computer Interaction and User Experience in Indonesia, CHIuXiD'18*. 18–22.

[25] K. Terajima, T. Komuro, and M. Ishikawa. 2009. Fast finger tracking system for in-air typing interface. In *CHI'09 Extended Abstracts on Human Factors in Computing Systems*. 3739–3744.

[26] C. T. Tolentino, A. Uy, and P. Naval. 2019. Air Drums: Playing Drums Using Computer Vision. In *International Symposium on Multimedia and Communication Technology*. IEEE, 1–6.

[27] L. Turchet, C. Fischione, G. Essl, D. Keller, and M. Barthet. 2018. Internet of Musical Things: Vision and Challenges. *IEEE Access* 6 (2018), 61994–62017.

[28] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C. L. Chang, and M. Grundmann. 2020. Mediapipe hands: On-device real-time hand tracking. *arXiv preprint arXiv:2006.10214* (2020).