Haptic Interaction with Guitar and Bass Virtual Strings

Haptic Interaction with Guitar and Bass Virtual Strings

Andrea Passalenti
HCI Laboratory
Department of Mathematics,
Computer Science and Physics
University of Udine
206, via delle Scienze, 33100 Udine
passalenti.andrea@spes.uniud.it

Federico Fontana
HCI Laboratory
Department of Mathematics,
Computer Science and Physics
University of Udine
206, via delle Scienze, 33100 Udine
federico.fontana@uniud.it

ABSTRACT

A multimodal simulation of instrumental virtual strings is proposed. The system presents two different scenes under the Unity3D software, respectively representing guitar and bass strings. Physical interaction is enabled by a Sensable Technologies Phantom™ Omni, a portable haptic device with six degrees of freedom. Thanks to this device, credible physically-modeled haptic cues are returned by the virtual strings. Audio and visual feedback are dealt with by the system, too. Participants in a pilot user test appreciated the simulation especially concerning the haptic component.

1. INTRODUCTION

Force feedback is an actively researched subject in HCI, for the wide set of contexts in which it can be applied. There exist many different types of commercial technologies for the production of force feedback, each with its own size, degrees of freedom (DoF), range of motion, maximum exerted force and so on. We used the Phantom™ Omni haptic device. Often such technologies are developed following the DIY culture [1], by hacking existing systems in order to improve their features with haptic feedback. There are several application fields in which force feedback is desired: gaming, surgery, 3D modeling and, especially in recent years, music performance.

Musical haptics [2] can occur either *physically*, when a robotic arm interacts directly with a real instrument, or *virtually*, when the instrument is virtually synthesized through a software. We deal with the latter. Our project considers an interface developed with [1] Unity3D, displaying a guitar string set or a bass string set. The interaction with the virtual strings takes place by pointing over them through a virtual plectrum, whose movement in the screen is guided by the position of the robotic arm. The Phantom device is controlled within the Unity environment via the Haptic Plug-In For Unity made at the Digital Design Studio of the Glasgow School of Art [3]. Thanks to this plug-in, the device can be set to provide tactile feedback to the user containing positional information about the notes as well as string resistance, stiffness and vibration similar to those felt when one plays a string instrument.

[1] unity3d.com

Figure 1. Phantom™ Omni device by SensAble.

The Phantom Device has already been used successfully in interactive sound synthesis. In the Haptic Theremin [4] the physical position of the robotic arm in the 3D space was mapped into loudness (y direction), pitch (x direction) and distortion (z direction). In this case the C programming language *OpenHaptics* API was interfaced with the OpenGL environment, to create a custom graphic and haptic application.

2. HAPTIC INTERFACE AND INTEGRATION WITH UNITY

Phantom is able to return force and vibratory feedback cues fitting well with certain musical interactions [5]. With the exception of percussions and key-based controls, musical instruments in fact react smoothly to user actions, exerting forces that this device is able to reproduce. Its relatively low-power servo-motors have the further advantage to be especially silent, hence producing low auditory interference with the rest of the interface. For this reason, even if marketed since 1994 Phantom still represents an interesting choice for sonic interaction designers. Specifically for our purpose of recreating the tactile feel of guitar string

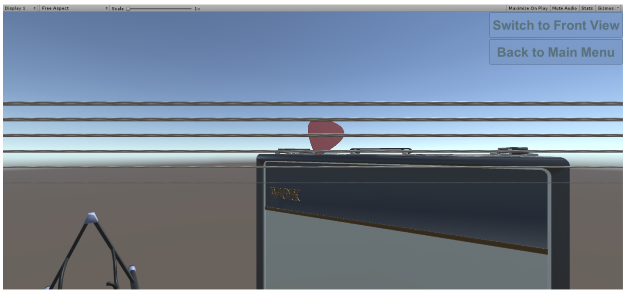

Figure 2. Guitar strings scene.

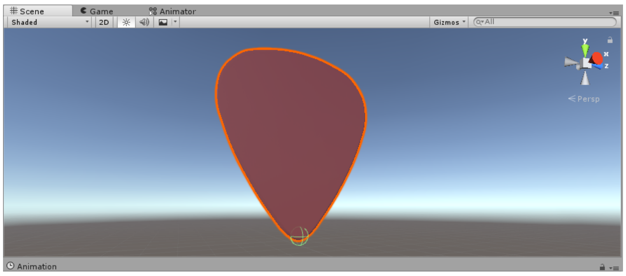

Figure 3. Spherical tip modeling plectrum rendering.

textures, when its servo-mechanisms were driven to reproduce this type of metal surface they generated a soft scraping sound as a by-product, providing immediate sense that a frictional sonic event was taking place. This sound could be directly used as part of the auditory display instead of disturbing it.

2.1 Interface

The interface includes two distinct scenes, showing respectively a set of guitar and bass strings. The user viewpoint is set to be behind the instrument, hence with the plectrum behind the string set as shown in Figure 2.

The user interacts with the strings through a virtual plectrum, whose edge position corresponds to the tip of the stylus that is attached to the robotic arm. The interaction hence is 'pen-like', as afforded by the Phantom device. It is physically modeled as a single point located in the center of a spherical shape object, modeling the tip of the virtual plectrum (Figure 3). The spherical shape in fact allows for establishing an invariant distance between its center (that is, the contact point) and the surface.

If the user pulls the stylus then the plectrum eventually interacts with one or more strings. There are different kinds of interactions:

- *Pick*, including two possible actions:
  - *Contact*: contact between plectrum and string;
  - *Release*: end of the contact, release of the string.
- *Scrape*, composed of two possible actions:
  - *Contact*: first contact between plectrum and string;
  - *Move*: longitudinal movements along the string without release.

After string release, a pre-recorded string sound is reproduced.

3. SENSORY MODALITIES AND FEEDBACK

The sensory modalities involved in the interaction are three: tactile, auditory, and visual. The corresponding feedback events are managed by C# scripts run by Unity3D.

3.1 Tactile modality

In our project we pay particular attention to the management of tactile feedback. *Haptic plug-in*, developed in [3], is fundamental to return a correct haptic perception, that is the ability to explore the surface of an object trough an active contact with it during tactile perception. The plug-in in fact provides a set of haptic parameters that control the Phantom device directly. All haptic functionalities are managed by ASimpleHapticPlugin.dll, a library that controls the signals received from and sent to the Phantom device via firewire communication. This library exposes such controls in form of methods that can be invoked from a C# script run by Unity3D.

The library features interaction with the virtual objects (simple contact, manipulation in 3D space and other custom interactions) and the feedback retuned in case of contact (stiffness, damping, static and dynamic friction) between such objects, providing magnitude data and complete information about the position of the Phantom during the contact. Data range between 0 and 1, corresponding to magnitudes of each contact property that is managed by the plug-in. Further accessibility to ASimpleHapticPlugin.dll is unfortunately not allowed, nor sufficient documentation seems to exist to gain deeper control of the interactions through this library, for instance by manipulating the contact models and, hence, tactile and sonic feedback they could consequently generate.

A study on real strings behavior has been conducted to correctly tune these parameters, considering the response obtained by the interaction between the plectrum and the strings. In fact, every string has its own characteristics of gauge and wounding. String gauges are given with respect to the highest (and thinnest) string of the set. We have considered 0.09 mm gauge for guitar and 0.50 (wounded) mm for the bass (Figure 1, Figure 2). On guitars, wounded strings are usually the thickest, that also return a higher friction level with respect to the unwounded string. On the other hand, unwounded strings are the thinnest (even if the most stretched) and they results softer during picking. The bass strings follows the same physical characteristics, with the only difference that all strings are wounded.

A rough feedback during scraping has been obtained by setting the values of *static* and *dynamic* friction parameters. Static friction help simulate breakaway force, conversely dynamic friction produces roughness feedback during scraping. Figure 1 and Figure 2 show that friction parameters decrease their values according to string gauge and wounding.

Rigidity of a string during contact has been obtained by changing *stiffness* and *damping* parameters. This time, *stiffness* controls how hard a string surface is, and *damping* reduces the "springiness" of the string. Their values are directly proportionals to the string gauge, obtaining higher values for thickest strings and lower values for the thinnest ones, as shown in Table 1 and Table 2.

This approach proved effective. A guitar and a bass player first learned to operate on the parameters in Unity3D, then they started to calibrate the haptic properties of the virtual strings by repeatedly testing the result on the Phantom device side by side with a real instrument, according to a trial-and-error calibration process. In the end they declared to have obtained good matching between the feedback from the haptic device and the musical instrument.

Guitar

Gauge	Stiffness	Damping	S. Frict.	D. Frict.
E (0.42w)	0.26	0.21	0.42	0.48
A (0.32w)	0.19	0.17	0.28	0.35
D (0.24w)	0.15	0.12	0.10	0.15
G (0.16)	0.07	0.055	0.02	0.028
B (0.11)	0.05	0.03	0.015	0.02
e (0.09)	0.032	0.021	0.01	0.013

Table 1. Ernie Ball Super Slinky (0.09) - Dunlop Torex 0.88mm

Bass

Gauge	Stiffness	Damping	S. Frict.	D. Frict.
E (1.05w)	0.65	0.56	0.9	1
A (0.85w)	0.55	0.48	0.79	0.91
D (0.70w)	0.48	0.42	0.61	0.73
G (0.50w)	0.39	0.35	0.47	0.55

Table 2. Ernie Ball Regular Slinky (0.50) - Dunlop 449 Max Grip 0.88mm

3.2 Auditory modality

Auditory feedback was the result of superimposing sound reproductions from Unity3D to real sounds from the Phantom device. Concerning the reproduction, we considered a set of open strings with standard tuning: E, A, D, G, B, e for guitar and E, A, G, D for bass. Audio samples were recorded by the first author across a large campaign, covering a comprehensive set of sounds resulting from playing a guitar with picking style.

String percussion generally returns a sound, whose loudness is proportional to the dynamics involved. The sound then fades until silence is reached. In particular, the behavior with high dynamics corresponds to a loud sound even if a little sharp note is involved. This sound, initially out of tune, tunes back when a smaller dynamic is reached. We considered two sets of samples for every string: the former consisting of high dynamic (little sharp) notes, and the latter considering normal dynamics (tuned) notes. Both such sets were associated to open string objects. Whenever a release event occurs, Unity3D calls a C# script that maps dynamics into speed, and updates an *audio source* clip assigning a little sharp note sample if the speed of release exceeds a given threshold, otherwise it assigns a tuned note sample.

In parallel, scraping sounds were returned by the Phantom device that, thanks to the noise naturally coming from its servo-motors during contact events, ensured an auditory feedback very similar to the real one. Initially the idea was to manage the whole auditory display from Unity3D, but later the real sound from the servo-motors turned out to have a better appeal than any sample-based reproduction, if not because reproducing auditory scraping events in perfect synchronization with the corresponding haptic feedback proved to be difficult. A scraping event takes place on wounded strings because of their rough surface, happening when the plectrum rubs on a string through its edge. A consistent auditory counterpart should consist of a small particle of a scraping sound, being played repeatedly during the entire interaction event meanwhile varying its pitch and its reproduction speed proportionally to the action. This process results almost impracticable if compared to the simplicity and realism of the auditory feedback coming from the motors.

3.3 Visual modality

This modality was completely managed from Unity3D. It consists of the representation of the moving plectrum in the 3D space, along with the vibrating strings after a picking event. The plectrum simply follows the position of the handled stylus. In particular, for every position in the 3D space it can rotate on its x, y, z axes with respect to the plectrum tip. To allow for a simpler interaction, the plectrum movements were limited to the plane of the strings in ways that it could not be moved beyond their position.

Every string was represented as a cylinder object. Such cylinders were positioned horizontally, with distances identical to those existing in a real lead and bass guitar. String vibrations were resolved using Unity3D *animations*, in a way to produce visual oscillations proportional to the dynamics happening during a release step.

First, an entire set of animations was created for every string. Each animation represents a vibration intensity, and is composed by two displacements of the cylinder with respect of the original position: one up, and one down. The sets of *animations* were managed using an *animator* object, that is an interface able to control the animation system. In particular, we associated an animator object to every string, able to create a sequence between its animations from the widest to the narrowest, hence creating a visual effect of decreasing vibration until stop. The sequencing of the animations was refined until providing a visual sense of continuity across the steps. Dynamics referred to the initial speed accumulated during the release event, and consequently they started from different points of the animation sequence proportionally to this speed.

A vibration could stop either by natural fade out, or if another contact event between the plectrum and the string happened. This action returned also a short ringing sound just before the stop.

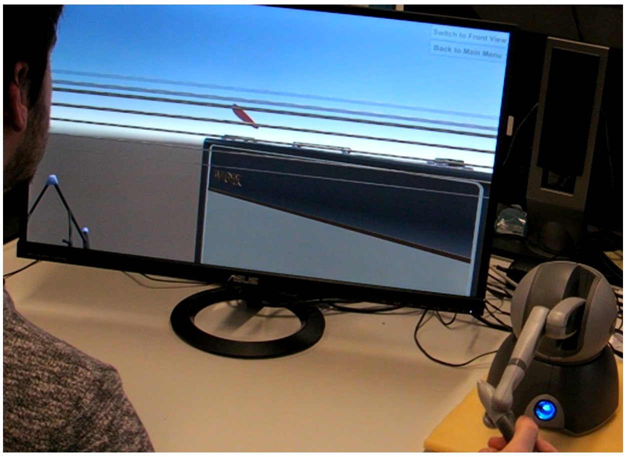

Figure 4. Testing session at the HCI Lab.

An approach based on animations represents a simple and fast method to reach our purpose. Unfortunately, it does not allow to reach a sufficient level of realism. A viable improvement would operate on the string transparency during vibration, which can be obtained by varying this parameter proportionally to the oscillation amplitude. Also in this case, however, the result is not completely satisfactory as during the oscillation the entire cylinder moves up and down as a stiff bar. Another idea is to replace the whole cylinder object with a 3D string model object, able to bend its structure during the vibration as it happens in the real world. This solution is much more difficult to develop but, if computationally not too demanding, would give better results.

4. TEST AND RESULTS

In this section we present the results of a preliminary test we conducted on a group of 7 participants, with different age, music experience and knowledge of the technology. The test included two sessions. During the first session participants were asked to play with the real musical instrument—indeed the instrument that was used to calibrate the simulation (see Figure 1 and Figure 2)—using a plectrum. During the second session they were invited to try the application, by holding the Phantom stylus and interacting with the virtual strings.

Participants were left free to get accustomed with the positioning and holding of the stylus. After they felt comfortable with the computer interface, the individual session was stopped and they were asked to compile a questionnaire. Questions were divided in three sections: two of them aimed at comparing the musical interface with the real instrument concerning the auditory/haptic and visual modality; finally, the third section collected opinions about some design aspects of the interface.

Regarding the auditory/haptic feedback, participants were asked to rate *roughness*, *longitudinal motion friction* (i.e., resistance opposed to the plectrum movements), *sound friction* (i.e., sounds produced by servomotor) and *string resistance* (i.e., string stiffness and damping). Figure 5 shows good scores for the auditory/haptic responses, apart from

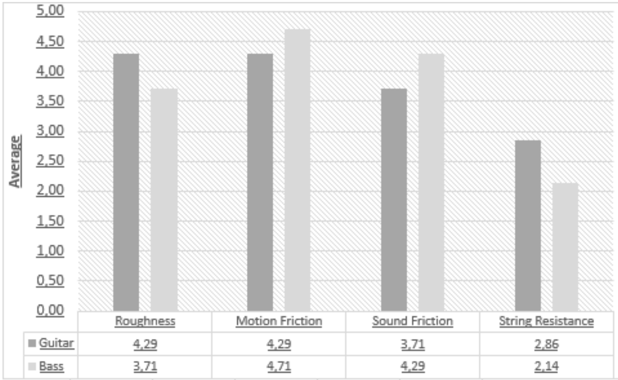

Figure 5. Results from haptic evaluation.

string resistance that results to be below such scores. The reason for scoring this attribute lower might depend on the stylus, which is different from a real plectrum concerning its shape, weight and affordable gestures.

This aspect is underlined in Figure 9 and Figure 10, where it can be noticed that the stylus is negatively rated by the majority of the users. Figure 6 illustrates the values obtained for each string, showing how the middle strings are less appreciated than the edge strings for the guitar; for the bass instead, only the last string was less appreciated.

Figure 7 shows scores resulting from visual evaluation. In this case the users were asked to evaluate *oscillation movements*, *oscillation range* and *oscillation time*. Concerning *oscillation movements* and *oscillation range*, the bass guitar was scored better than the lead guitar. This difference can be explained almost certainly by our perception of string bending. In fact bass strings, having a thicker gauge, do not need a bending effect as pronounced as do guitar strings. For this reason, the dynamic seems to be more realistic for the bass. On the other hand, the oscillation had a more realistic frequency for the guitar, due to the limited amplitude. Indeed, in the case of the guitar the oscillation frequency is greater than the refresh rate, hence creating a transparency effect in the animation similar to the real one.

Concerning the participants' opinions about implementation choices, quite different results were found. Participants spent little time to get used to the Phantom stylus, although finding it sometimes uncomfortable and not much likely to the original. This fact must not surprise, being that stylus much different from a plectrum.

Overall, participants scored the haptic feedback better than the visual one. This result reflected our expectation, that the haptic design of the proposed virtual instrument was definitely more advanced and interesting to be experienced than its graphic design, which conversely should need to be improved in several aspects.

5. CONCLUSIONS

In this paper we have presented a multimodal interface simulating guitar and bass strings. The project was born with the idea to assess the potential of Phantom devices to take care of the haptic modality while designing a vir-

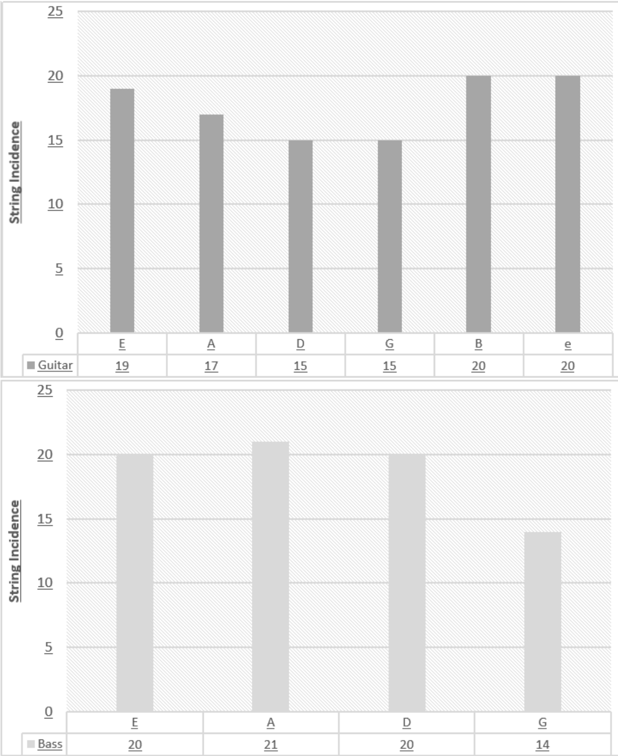

Figure 6. Individual contribution of scores from guitar and bass strings on haptic evaluation.

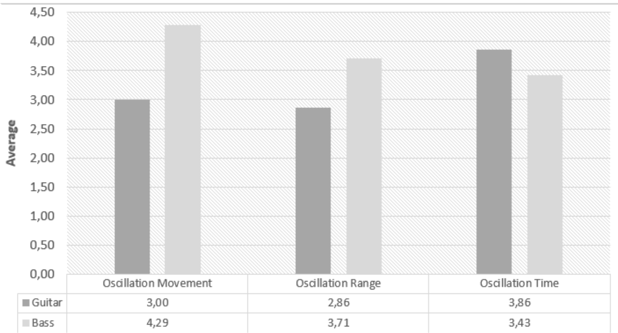

Figure 7. Results from visual evaluation.

tual guitar. Thanks to its low noise, reduced size, and accurate response of the servo-motors, this device simulates an effective plectrum-string interaction. Preliminary user tests prove that good results have been reached concerning this modality, especially when representing string roughness and motion friction that were furthermore supported by realistic noise coming from the motors when engaged in rendering these attributes. This support is even more significant because it prevented to set up a complicate, and computationally demanding sound reproduction of auditory events of scraping.

Weak points of our implementation regard its visual feedback and the stylus attributes. The latter differ too much from those of a plectrum. The visual display could be improved using more sophisticate 3D models, in order to return the string bending during its oscillation, also exploiting transparency effects provided by Unity3D.

Nowadays virtual reality gives a wide set of ideas for

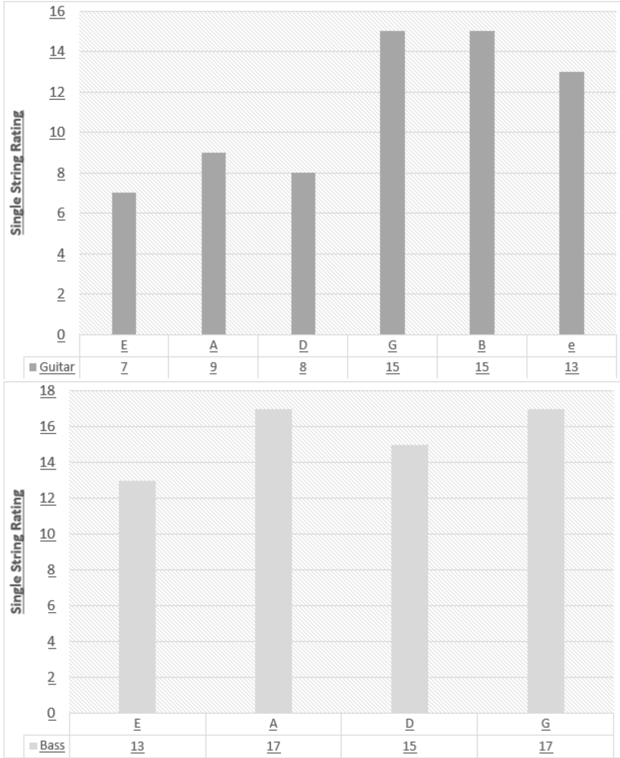

Figure 8. Individual contribution of scores from guitar and bass strings on visual evaluation.

expanding the musical vocabulary of the musical instruments [6]. In parallel, haptic technology can assist musicians in forming a rich gestural vocabulary [7]. Our aim for the near future is to expand this project, solving its weak points and adding the possibility to play chords through an external keyboard, in ways to propose the interface as part of a new musical instrument.

6. REFERENCES

[1] S. O'Modhrain, "Playing by feel: Incorporating haptic feedback into computer-based musical instruments," Ph.D. dissertation, Stanford University, 2000.

[2] S. Papetti and C. Saitis, Eds., *Musical Haptics*. Berlin: Springer, May 2018, in press.

[3] M. Poyade, M. Kargas, and V. Portela, *Haptic Plug-in for Unity*, Digital Design Studio, Glasgow School of Art, Glasgow, Scotland, June 2014. [Online]. Available: http://radar.gsa.ac.uk/3575/6/Haptics_Plug_READ_ME.pdf

[4] W. P. Moss and B. Cunitz, "Haptic Theremin: Developing a haptic musical controller using the Sensable Phantom Omni," in *Proc. of the International Computer Music Conference (ICMC)*, Barcelona, Spain, September 2005.

[5] A. J. Silva, O. A. D. Ramirez, V. P. Vega, and J. P. O. Oliver, "Phantom Omni haptic device: Kinematic and

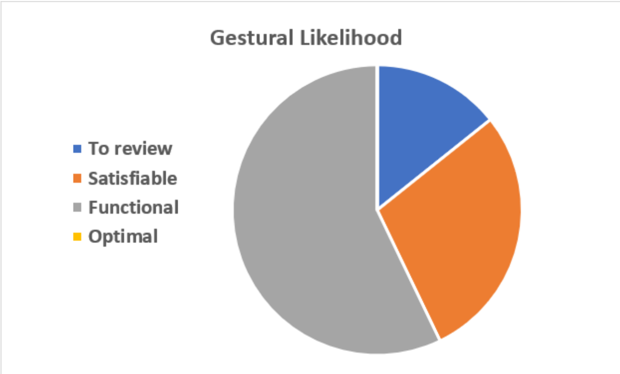

Figure 9. Similarity between gestures in the real and virtual instrument.

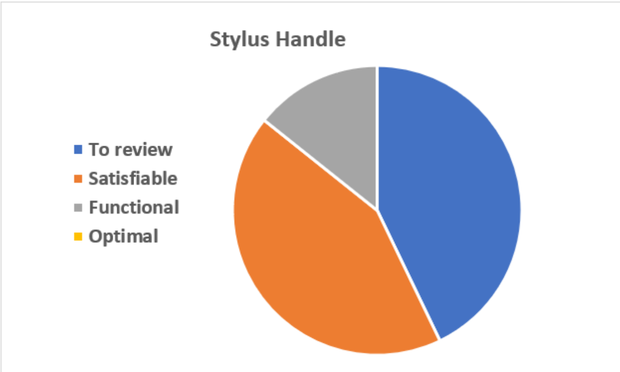

Figure 10. Similarity between handling in the real and virtual instrument.

manipulability," in *Proc. Electronics, Robotics and Automotive Mechanics Conference, 2009. CERMA'09.* IEEE, September 2009, pp. 193–198.

[6] A. McPherson, "Touchkeys: Capacitive multi-touch sensing on a physical keyboard," in *Proc. New Interfaces for Musical Expression (NIME 2012)*, Ann Arbor, US, May 2012.

[7] E. Berdahl, G. Niemeyer, and J. O. Smith, "Using haptics to assist performers in making gestures to a musical instrument," in *Proc. New Interfaces for Musical Expression (NIME 2009)*, Pittsburgh, PA, June 2009, pp. 177–182.