Glasstra: Exploring the Use of an Inconspicuous Head Mounted Display in a Live Technology-Mediated Music Performance

Ivica Ico Bukvic
Virginia Tech
School of Performing Arts
Institute for Creativity, Arts, and Technology
Blacksburg, VA, USA
ico@vt.edu

Spencer J. Lee
Virginia Tech
Department of Computer Science
Blacksburg, VA, USA
zamfir@vt.edu

ABSTRACT

The following paper explores the Inconspicuous Head-Mounted Display within the context of a live technology-mediated music performance. For this purpose in 2014 the authors have developed *Glasstra*, an Android/Google Glass networked display designed to project real-time orchestra status to the conductor, with the primary goal of minimizing the on-stage technology footprint and with it audience's potential distraction with technology. In preparation for its deployment in a real-world performance setting the team conducted a user study aimed to define relevant constraints of the Google Glass display. Based on the observed data, a conductor part from an existing laptop orchestra piece was retrofitted, thereby replacing the laptop with a Google Glass running *Glasstra* and a similarly inconspicuous forearm-mounted Wiimote controller. Below we present findings from the user study that have informed the design of the visual display, as well as multi-perspective observations from a series of real-world performances, including the designer, user, and the audience. We use findings to offer a new hypothesis, an inverse uncanny valley or what we refer to as uncanny mountain pertaining to audience's potential distraction with the technology within the context of a live technology-mediated music performance as a function of minimizing on-stage technological footprint.

Author Keywords

Inconspicuous Head-Mounted Display, Uncanny Mountain, Google Glass, Glasstra, Laptop Orchestra, Android, User study, Performance

ACM Classification

H.5.2 [Information Interfaces and Presentation] User Interfaces–Evaluation/methodology, H.5.2 [Information Interfaces and Presentation] User Interfaces–User-centered design, H.5.5 [Information Interfaces and Presentation] Sound and Music Computing–Systems

1. INTRODUCTION

One of the challenges of the innovative technology-mediated music performances commonly nurtured in the NIME community is the inevitable demo syndrome [1]. In addition to the one-off implementations that may be difficult, if not impossible to reproduce by a third party, such performances have a tendency to distract from the content (e.g. music, physical presence and choreography, and/or audio-visual material) with their technological "wow" factor. It is not uncommon for an audience member to be thoroughly impressed by the innovative use of the technology. Yet, when asked to comment on the experience, their feedback almost exclusively focuses on the technology itself with little recollection of the actual content that is arguably the primary reason for the design of a NIME.

While such a technology will over time lose its "wow" luster and with it the ability to distract the audience from the content it is designed to deliver, waiting for such a time to pursue the profound artistic depth is cumbersome. On the one hand, those who persist at pursuing greater depth by sticking to the same technology, may over time end up being perceived as not being on the cutting edge, making it potentially difficult to publish on the newly uncovered nuances of the well-established technology. On the other hand, there will always be newer technologies and scholars who will be eager to pursue them within the context of NIME, as well as other transdisciplinary domains. This realization promotes a seemingly endless demo syndrome cycle in which scholars are doomed to jump onto whatever the latest technology may bring without having the opportunity to stop, reflect, and build the necessary depth. It is also worth noting, this challenge is not unique to the NIME community. Rather, it is inherent to all domains focusing on the exploration and integration of the new technologies.

1.1 Motivation

One of the primary motivations of this paper is exploring ways to minimize the on-stage technological footprint within the context of live technology-mediated music performance. By removing such observable on-stage technological presence with a potential "wow" factor, we envision audience members being in a better position to redirect their attention away from technology and towards the content. In addition, such a technological solution could allow for performer's improved mobility and freedom of motion.

The aforesaid aspirational goal undoubtedly requires a holistic approach to supplanting a whole array of the existing technologies with their more inconspicuous and flexible contemporary alternatives, and as such goes well beyond what we can tackle in a single paper. For this reason, here we focus on a subset of such potential alternatives, namely the head-mounted display (HMD) [2] as an alternative to a laptop display. When used in conjunction with an inconspicuous wireless controller an HMD has a potential to supplant a laptop and supporting controller infrastructure, arguably some of the most visible technologies commonly found on-stage. Given HMDs by default tend to be bulky and conspicuous, thereby potentially exacerbating the very problem this paper aims to address, hereby we propose a name Inconspicuous Head-Mounted Display (IHMD). IHMD describes a subset of HMDs that offer minimal technological footprint and are ideally imperceptible to the audience.

1.2 Context

Since its introduction in 2009, Linux Laptop Orchestra (L2Ork)'s mission has been exploring innovative ways to

NIME'17, May 15-19, 2017, Aalborg University Copenhagen, Denmark.

compose and perform technology-mediated live ensemble music [10]. With its focus on gesture and integration of Tai Chi choreography [11], L2Ork in particular seeks integration of wearable technologies that have a potential to minimize on-stage technological footprint and by doing so promote performers' freedom of motion, as well as ideally divert audience's attention away from technology and towards the content.

In technology-mediated music performance, particularly situations where NIMEs are incapable of independently providing adequate secondary (e.g. haptic) feedback, performers' eyes are all too often fixated onto the laptop screen. Despite extensive exploration of the haptic feedback using the built-in Nintendo Wiimote's [3] rumble functionality, L2Ork performers continue to exhibit overreliance on the visual feedback found on the laptop screens. This limits not only the scope of their gestures and consequently choreography, but also their ability to establish an eye contact with the conductor, each other, and the audience.

An IHMD has a potential to further L2Ork's mission, as well as address the aforesaid challenges by:

1) Decoupling visual data from a static laptop screen and allowing users greater freedom of motion whereby the information is readily available regardless of their location, head orientation, or body posture, and
2) Lowering technology footprint where entire laptop and the supporting music stand can be ostensibly replaced by a wearable device and thereby promoting eye contact between the ensemble members, as well as performers and the audience.

2. DESIGN AND IMPLEMENTATION

Google Glass (Glass) [4] is an Android-based wearable device with a monocular see-through optical head-mounted display (HMD) with a screen resolution of 640x360. Since its introduction in 2013, its usability was explored in various areas such as education [5], medicine [6], music [7], etc. Although today it is a defunct project in part because it was a costly prototype introduced ahead of its time, it inspired the HMD industry and numerous companies seeking to further explore it, including HoloLens [8], SmartEyeGlass [9], and Meta pro [10]. More importantly, its innovative push towards the inconspicuous form factor makes it arguably one of the first examples of a publicly available IHMD. Consequently, in order to assess the potential impact of an IHMD within the context of a live technology-mediated music performance we chose Glass as our research platform. The project implementation was split into multiple stages:

1) Implementing *Glasstra*, a lightweight network-enabled Android/Glass client;
2) A user study to identify Glass display limits within the context of a live music performance, and
3) Real-world testing.

3. GLASSTRA

To evaluate Google Glass as an IHMD in the context of a live ensemble music performance, there is a need for a lightweight networked Android/Glass client that could be driven remotely and reconfigured at runtime. Such an implementation would provide a universal solution satisfying needs for both stages two and three, including iterative improvements based on the study findings. Given L2Ork's primary software infrastructure is Pd-L2Ork [11], the client would need to be capable of communicating with Pd-L2Ork using the Pure-Data's FUDI protocol [12].

The client's primary purpose would be to dynamically display and update a collection of customizable widgets akin to iemgui objects onto the small Glass display. In this respect the proposed client is not unlike Pure-Data-centric PdDroidParty [13] and MobMuPlat [14], Mira [15], or the platform-agnostic TouchOSC [16], with one critical difference: *it needs to be designed specifically for Glass IHMD and its constraints, including limited computing power and display size*. In addition, akin to Max's Mira it would be need to be dependent on the remote server's networked FUDI packets responsible for dynamically displaying, updating, and erasing widgets. Another, more nuanced difference, in part inspired by the aspirational lightweight implementation and low CPU overhead, is its focus on displaying content, rather than directly interacting with it via built-in display sensors. In other words, at least in its initial iteration the proposed client would serve solely as a display of information, thereby offering a minimally intrusive wearable counterpart to a conventional laptop display.

To address the aforesaid whitespace, we created *Glasstra* free open source Android/Glass application using Glass Development Kit (GDK). *Glasstra* offers FUDI-compatible networked communication using either TCP or UDP packets. It constructs a blank canvas that can be populated with iemgui-like collection of widgets, including a bang, toggle, vertical and horizontal sliders, a graph, and a text box. Each object offers dynamic script-based customization, including alpha blending and color assignment. As a result, widgets can be easily used in ways that defy their original intent (e.g. using a toggle as a colored canvas, or creating a floating text with a transparent text box).

Widget	Attributes	Description (default)
All	x y (float)	On-screen position
	w h (float)	Width and height
	destroy	Destroy an object
	visible (0/1)	Toggle object visibility (1)
	fc1 (char ARGB)	Inner shape edge color (gray)
	fc2 (char ARGB)	Inner shape background color (dark gray)
	bc1 (char ARGB)	Outer shape edge color (light gray)
	bc2 (char ARGB)	Outer shape background color (light gray)
	hc (char ARGB)	Highlight color (red)
bang	Cooldown (int)	Bang cooldown in milliseconds (50)
toggle	on (0/1)	Toggle off/on (0)
vslider hslider	range (int int)	Set slider min and max range (0, 127)
graph	xpoints (int)	Set graph horizontal size (10)
	yrange (float)	Set graph's Y range (-1, 1)
	Set (float array)	Set graph values y1 y2 y3 etc.

Table 1. *Glasstra* commands and attributes.

3.1 *Glasstra* Script

In order for a laptop to communicate with *Glasstra* the data is sent using Pd-L2Ork's built-in `disis_netsend` and/or Pure-Data's native `netsend`, both of which utilize the FUDI protocol and offer TCP and UDP connectivity. To connect, the Glass needs to be on the same wireless network as the laptop. Upon starting *Glasstra*, the screen will display Glass' IP address and *Glasstra*'s hardwired port of 55555 and will remain displayed until it receives a successful connection from a FUDI-compliant client (e.g. a laptop running Pd-L2Ork, Pure-Data, Max, etc.). *Glasstra* port is intentionally hardwired to minimize runtime configuration, which in the case of Glass tends to be limited to swipes and taps on the touch strip found on the right side of the Glass. The connection remains active until the application is closed.

The current list of commands *Glasstra* understands is shown in Table 1. All parameters are space delimited. Each widget is assigned name at creation, so that it can be referenced later. When being created, widgets also require creation parameters that have no defaults, including position, width, and height.

Special name "all" is reserved to address all existing widgets regardless of their state. For instance:

```
toggle tog1 100 60 50 50
tog1 fc1 255 0 0 255
tog1 on 1
...
tog1 destroy
...
all destroy
```

The example above creates a toggle named tog1 at a location 100 60 with the width and height of 50 pixels. It is assigned a new inner shape edge color (the area that is displayed when the toggle is on), and is then set to on state. Later, the object is destroyed and removed from canvas, and eventually all remaining objects are removed from the canvas. All commands have an implicit semicolon at the end of each line that is appended by the FUDI-protocol-compliant Pd-L2Ork and Pure-Data objects and interpreted accordingly by the *Glasstra*.

3.2 Initial Testing and Optimization

During the design phase *Glasstra* was tested for performance, CPU overhead, and battery usage. Performance was assessed by rapidly populating the canvas with hundreds of widgets and observing any potential slowdown. *Glasstra* tends to perform well with up to 50 concurrent objects after which its responsiveness precipitously drops. This was well above the typical Glass design guidelines [17] and as such we did not anticipate it as being an issue. The client was also designed to utilize a minimal amount of CPU. At idle time the CPU usage was negligible, regardless whether there was already an active connection and/or widgets populating the canvas. As expected, the CPU usage rose with an increase in the network packets and consequently dynamic updates (e.g. creating, updating, and erasing widgets). Yet, this increase remained manageable until updates exceeded the Glass refresh rate and client's ability to display them.

Battery usage was the unanticipated limitation. Even when running idle, due to limited battery size Glass' display and active wireless connection tended to drain battery quite rapidly, leaving no more than an hour before the battery is completely drained and Glass shut off. While an increase in the network traffic and the CPU usage had an observable impact on the overall battery life, it proved minimal when compared to the wireless chipset and display's power needs. For this reason, during the user study, the Glass was connected to a portable battery pack.

4. USER STUDY

Following *Glasstra*'s design, implementation, and technical testing, we conducted a user study leveraging *Glasstra* to evaluate Glass as an IHMD. In particular, our focus was to identify critical factors to Glass' usability and their thresholds pertaining to the live on-stage gesture-based music performance, thereby matching L2Ork's specific needs.

4.1 Participant Overview

24 participants (11 women and 13 men) took part in the user study. One of them had past L2Ork performance experience. 100% of them owned and used at least one smart mobile device, and used their device(s) frequently. 21% (5) of them have used a Glass before. 75% (18) of them wore glasses or contact lenses.

4.1.1 Outliers

There was one outlier among the 24 participants. The participant could not see the Glass's display at all. We found out that Glass's display is not visible if it is not located at the center of, and also perpendicular to, user right eye's optical axis. The Glass's display was lifted upward due to the participant (outlier)'s lower position of ears. The outlier was identified during the pre-experiment training, and excluded from the data collection.

4.2 Design

The user study was designed to evaluate users' ability to detect the onset of a visual event and/or a widget state/value change by answering the following questions:

1) What are the thresholds of widget visibility as a function of size and position;
2) How users' performance (object recognition) is affected according to the environmental (lighting) conditions, including:
    a. Bright (737-5 Lux), mimicking on-stage lighting;
    b. Darker (14-5 Lux), mimicking a more intimate indirect lighting conditions, and
    c. Completely dark with Glass display completely covered with a 3D printed cover (0 Lux), or control.
3) How big widgets should be when user's attention is not on the Glass (widgets are positioned in users' peripheral vision), thus once again mimicking typical music performance conditions.

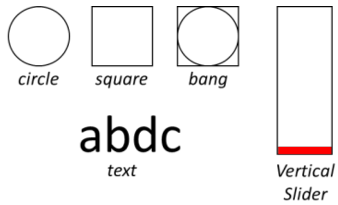

Figure 1. *Glasstra* widgets employed in the study.

The first objective was to determine the minimum sizes of objects to be displayed using *Glasstra*. As depicted in Figure 1, we used five basic iemgui-like objects: circle or a bang object without a frame, square or a toggle, bang or a combination of a circle and a square, vertical slider, and text. The bang object was also used to measure the minimum perceptible frequency of its flashing effect. The second objective was to ascertain any significant difference in perceptibility of *Glasstra*'s widgets displayed on the Glass under varying lighting conditions. The third objective focused on identifying proper notification strategies when user's primary attention is not on the Glass, something that is particularly common in gesture- and choreography-driven music performance.

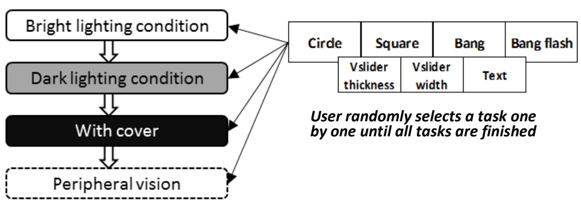

Figure 2. User study procedure.

4.3 Procedure

The user study was conducted in four phases, each with seven tasks as seen in Figure 2. In each phase, each task was in random order until all tasks were completed. For circle, square, bang, and text tasks, each respective widget was displayed starting with a size of 0 pixels (thereby being essentially invisible). The size was then increased over time by 1 pixel until the participant could clearly see and identify the object on the Glass display at which point they were instructed to say 'stop' which denoted the completion of the said test. For bang flash, vertical slider thickness and width tasks, initially each object was displayed populating most of the available screen, excluding the padding area, as per Google Glass design guidelines [17], while keeping default proportions. Each participant was asked to say 'stop' when they noticed any change to the widget at which point they were asked to describe the said change. All tasks were measured in pixel sizes except for the bang flash task that was measured in milliseconds. For the last, peripheral vision phase, each participant was asked to keep their eyes focused on the unrelated content on laptop's display positioned directly in front of them, and was instructed to say 'stop' when they noticed or felt any changes to the object shown on the Glass display without looking directly at it.

4.4 Results

Table 2 shows the minimum values of each task reflected either in pixel sizes or milliseconds.

	Circle	Square	Bang	Bang flash (ms)	Vslider thickness	Vslider Width change	Text
Bright	28	20	29	1	12	17	25
Dark	30	24	23	2	13	12	27
Cover	22	22	28	2	10	14	28
Peripheral	132	158	113	66	127	98	116

Table 2. The minimum values (in pixels or milliseconds) of each user study phase (vertical) and task (horizontal).

4.4.1 Analysis

We performed ANOVA analysis after normalizing the data as follows:

1) Convert data to proportions (divide by 360, or the vertical resolution of the Glass display);
2) Average them by group (circle, square, bang, text as one group), and
3) Perform logit transformation (Y*i* = log (P*i*/(1-P*i*)) in order to covert bounded data to unbounded data.

LSMeans Differences Tukey HSD

α= 0.050 Q= 2.61881

LSMean[j]

LSMean[i]	Mean[i]-Mean[j] / Std Err Dif / Lower CL Dif / Upper CL Dif	Bright	Cover	Dark	Peripheral
Bright	Mean[i]-Mean[j]	0	0.0313	-0.0136	-0.3597
	Std Err Dif	0	0.0604	0.0604	0.0604
	Lower CL Dif	0	-0.1269	-0.1718	-0.5179
	Upper CL Dif	0	0.18948	0.14455	-0.2015
Cover	Mean[i]-Mean[j]	-0.0313	0	-0.0449	-0.391
	Std Err Dif	0.0604	0	0.0604	0.0604
	Lower CL Dif	-0.1895	0	-0.2031	-0.5492
	Upper CL Dif	0.12688	0	0.11325	-0.2328
Dark	Mean[i]-Mean[j]	0.01363	0.04493	0	-0.3461
	Std Err Dif	0.0604	0.0604	0	0.0604
	Lower CL Dif	-0.1445	-0.1132	0	-0.5043
	Upper CL Dif	0.17181	0.20311	0	-0.1879
Peripheral	Mean[i]-Mean[j]	0.35973	0.39103	0.34609	0
	Std Err Dif	0.0604	0.0604	0.0604	0
	Lower CL Dif	0.20155	0.23285	0.18791	0
	Upper CL Dif	0.51791	0.54921	0.50427	0

Level			Least Sq Mean
Peripheral	A		-1.143927
Dark		B	-1.490020
Bright		B	-1.503652
Cover		B	-1.534952

Levels not connected by same letter are significantly different.

Tables 3a (left) and 3b (right). ANOVA results by JMP.

Tables 3a and 3b show the results of the statistical analysis performed by the JMP statistical discovery software [13] after consultation with Virginia Tech's Laboratory for Interdisciplinary Statistical Analysis (LISA).

Table 3a indicates that there's significant difference (red color) in peripheral vision phase, thus affirming a well-known human factor that "the peripheral vision is less acute than the foveal vision" [14]. Given Glass' display position is by design located within user's peripheral vision and that its use as an IHMD in a live motion- and gesture-centric music performance is likely to rely at least in part on the same, the peripheral test data set was chosen as the minimal size threshold for widget representation. This ensured the widgets employed in the real-world testing would be ideally perceptible both in direct attention and peripheral vision conditions. The same also negated any potential performance concern as observed in 3.1, as there simply was not enough screen area to display more than dozen concurrent widgets and ensuring state changes would remain perceptible to the user.

The analysis results also show there's no statistically significant difference in users' widget recognition performance. Table 3b further affirms there is no statistically significant difference in object recognition according to the different lighting conditions (Bright, Dark, and with Cover).

According to the answers of the post-study questionnaire, 71% (17 participants) of them experienced discomfort such as eye strain or headache while wearing the Google Glass, and 67% (16 participants) of them disagreed that Google Glass was comfortable to wear, while 71% of them agreed that the current setup was easy to use. At least a part of the observed fatigue can be attributed to the study's mundane nature and consequently duration.

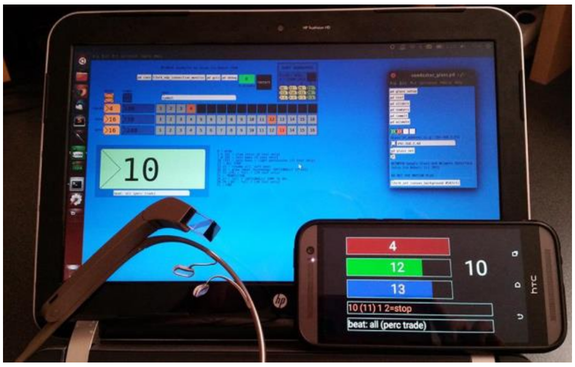

Figure 3. Pd-L2Ork musical notation displayed on the Google Glass using *Glasstra* (mobile phone on this photo mirrors what is on the Google Glass display).

5. REAL-WORLD TESTING

Based on the study findings we retrofitted a conductor part of an existing L2Ork composition titled *Between* by moving conductor's laptop off-stage and replacing the laptop display with the Glass IHMD running *Glasstra* that was wirelessly connected to conductor's laptop (Fig.3). Given the piece called for conductor to control sections using a limited set of traditional conducting techniques and system cues via a laptop keyboard, the IHMD was coupled with a Nintendo Wiimote that was strapped onto conductor's right forearm, therefore enabling the use of a subtle haptic feedback. The Nunchuk was connected to the Wiimote and inconspicuously placed in conductor's right hand, allowing them to cycle between sections using the Nunchuk's joystick and activate them using Nunchuk's Z button.

This piece was chosen because it offers a steady pulse, tight sync between the parts—each with its own tempo and meter, and accessible aesthetics, making possible errors clear to an untrained ear. In addition, the conductor is assigned a critical role in ensuring the piece is performed correctly and in sync, while concurrently being in charge of varying the final structure, thus minimizing the chance of performers learning the piece by heart and ignoring the conductor cues. While any part could be ostensibly retrofitted to use the Glass IHMD instead of a laptop screen, given the limited access to Glass hardware, we opted to retrofit only the conductor's part.

Conductor's ensuing setup required no adjacent hardware, allowing for the conductor to stand and move freely anywhere inside the performance space, including the audience. Similarly, this implementation allowed the ensemble members to be spread around the performance space without preventing the conductor from maintaining a peripheral view of the projected information regardless of their position or orientation. The premiere that took place in Virginia Tech Cube's as part of the SEAMUS 2015 conference called for performers to be spread all around the audience and on different elevations using Cube's catwalks, so as to enhance the ensemble's spatial potential.

5.1 Interface

The resulting conductor's *Glasstra* interface is shown in Figure 4. In addition to default uses of various widgets, it reflects the extended use of their properties to achieve functionality beyond their original design intent. For instance, in a screen populated by a number of widgets, it has proven necessary to provide a screen-wide rectangle (e.g. a toggle or a slider) as a color-changing background whenever a section change was invoked to maximize its impact on conductor's peripheral vision. This was particularly important considering the conductor had to also maintain eye contact with the performers to instill confidence and optimize ensemble's synchronization. *Glasstra*'s stacking order was further explored by juxtaposing alphanumerical information on top of visual sliders designed to reflect location of each of the work's three choirs within their respective pattern and tempos. The large number on the right reflected the current section. Finally, the two boxes at the bottom served as cheat sheets, helping conductor remember what section they were in and what section they needed to transition to, as well as what supplemental information was distributed to various choirs.

Figure 4. *Glasstra*'s conductor interface for the L2Ork's *Between* composition depicting a 3-step transition cue.

5.2 Feedback

Below we provide observations from three different perspectives: designer, user/conductor and the ensemble, and audience. Within the context of this study, Bukvic served both as the user and the conductor.

5.2.1 Designer

A live performance that thrives on a tight sync between parts requires changes to the visual display and its widgets to be near instantaneous. While in technical tests conducted during the *Glasstra*'s initial implementation there was no observable delay, this was likely in good part due to a relatively simple setup—no test required more than one widget to be concurrently displayed or updated. What we learned through a series of tests during the real-world design phase using a combination of different Glass hardware and wireless routers is that, in addition to the inherently unpredictable time jitter of wireless data packets, certain routers and Glass variants were much better at timely handling of anything from simple pings to *Glasstra*'s FUDI-formatted network packets. The consistent behavior between the pings and Glass hardware confirmed that the problem was not associated with *Glasstra*. This meant that not all Glass hardware was made equal. Indeed, different iterations used different wireless chipsets, often resulting in unworkable latencies. When used in conjunction with TCP packets, they could easily exceed two seconds, while the UDP packets could at times arrive out of place, resulting in ignored and/or mangled instructions. These observations are essentially limitations of wireless communication that were further amplified by Glass' low power hardware design. Upon identifying optimal Glass hardware iteration and a high-end multi-antenna wireless router with a beamforming capacity the problem was minimized consistently to sub-25ms latencies and no observable dropped packets. This meant the conductor still had to rely primarily on the internal sense of pulse when it came to accurately timing individual sections, using the IHMD to monitor sync between the parts and anticipate triggering different sections, as well as to keep track of the overall progress through the work's structure.

Apart from the aforesaid unforeseen technical challenges, the design was a straightforward iterative process. Perhaps the most notable limitation was having to rely on the scripted language, rather than a graphical editor that would've helped streamline the overall design process.

5.2.2 User/Conductor and the Ensemble

With the design complete, the *Glasstra* interface was used through a series of rehearsals and eventually in three real-world performances. A relatively short battery life at times proved cumbersome and during extended rehearsals, it was necessary to rely on a portable battery pack. In performances, the Glass was left to charge until needed, which proved more than adequate for an approximately 10-minute long piece that with 70+% of battery life remaining.

Setup, although streamlined, still required a transition time to adjust and properly align the display, particularly in situations where the user wore conventional prescription glasses. It is worth noting the Glass can be coupled with prescription lenses, thereby making this problem largely a non-issue. From a conductor's perspective, despite the low CPU footprint Glass tended to get noticeably warm over time, making it less comfortable to wear.

Perhaps one of the greatest concerns was the wandering or rolling eye effect that was necessary to read finer details (e.g. a section number, or the section cheat sheet) by focusing eyes onto the display located in the top-right corner. This eye motion from performers' perspective was at times misinterpreted as a momentary gaze in a different direction, potentially sending a mixed message, such as a rolling eye as a sign of annoyance or dissatisfaction. This was, however, alleviated through repeat exposure and practice and performers quickly adjusted to the anomaly, while the conductor refined their head orientation during momentary gazes onto the Glass' display to minimize potential confusion.

5.2.3 Audience

Perhaps one of the most compelling findings of this project was its impact on the audience. Even though all of the performances required the conductor to be positioned in the center of the performance space surrounded by the audience members, sometimes seated as close as only a few feet away from the conductor, most of the audience members failed to notice the Wiimote and the Nunchuk, many failed to notice Glass, and a few failed to notice both. This may be in part because the piece called for performer distribution around the audience, leaving ample visual cues for the audience members to direct their attention towards. Following the performance the ensemble received a number of questions pertaining to how the piece was conducted and how was the sync ensured between different sections. Similarly, those who only noticed the Glass inquired afterwards as to what was its purpose, being unaware of conductor's ability to issue commands through the hidden Nunchuk. No audience member reported noticing wandering or rolling eye effect.

5.3 Discussion and Conclusions

Based on the audience feedback, the project managed to significantly minimize the overall technological footprint for the conductor's part, effectively rendering the entire performance space traversable and thereby achieving a significant progress on the first aspirational goal. Yet ironically, it has also further exacerbated audience's curiosity about the increasingly inconspicuous use of the technology. Although rendering the technology near invisible is deemed a success, and a number of audience members favorably commented on the music experience, it is unclear whether the project has made any progress towards minimizing the technological distraction. It appears the effect of pursuing the ever-smaller technological footprint towards as of yet unattainable goal of making technology completely invisible exhibits a pattern inverse to that of uncanny valley [18] or what we hereby refer to as the *uncanny mountain*: as we approach the ultimate goal of removing all visible traces of the supporting technology, we are facing a seemingly insurmountable surge in audience's interest in and infatuation with technology over that of the content the technology is designed to deliver. In other words, as we approach the invisible, the audience interest is increasingly fixated on how rather than what. If the way we approach creative work that lacks the technological "wow" factor is any indicator, then perhaps attaining the absolute technological transparency and by doing so traversing the aforesaid uncanny mountain will finally bring about the undivided and permanent attention to the content, rather than technology.

6. FUTURE WORK

While we envision utilizing *Glasstra* both as a performer and a conductor interface, to date its utilization has been restricted to the conductor role only. This has been in part due to restrictive cost and the lack of access to a necessary quantity of the Glass hardware, as well as a desire to study its potential as an IHMD in a technology-mediated live music ensemble performance in a controlled and manageable capacity.

As part of this study we consciously avoided the use of Glass' built-in sensors, thereby focusing solely on the study of its IHMD potential. This is something that needs to be explored in future iterations. However, given the Google Glass is now effectively obsolete, the follow-up studies will require alternative hardware, ideally also minimizing the overall cost. Possible solutions include minimal displays powered by a wearable Raspberry Pi or a similar low power microcomputer. Current findings also warrant further studies towards seeking solutions with longer lasting battery life.

The three performances to date featuring *Glasstra* have proven a success with no observable technical difficulties and were received with overwhelmingly positive audience and user feedback. The current feedback, however, does not clarify whether positive impressions are technology- or content-centric and further studies are warranted to explore the effects of the ongoing effort to minimize the technological footprint and the newly hypothesized uncanny mountain.

Lastly, given *Glasstra*'s ability to render a dynamic visual display broadcast over network, it has a potential to prove its usefulness in rapid prototyping, and consequently a much broader array of potential use scenarios that go well beyond the laptop ensemble paradigm. We have already seen early examples of such extended uses in the research in HMD-based interfaces conducted by the Industrial and Systems Engineering Department at Virginia Tech.

7. ACKNOWLEDGMENTS

The authors hereby acknowledge Virginia Tech's Institute for Creativity, Arts, and Technology (ICAT) for the project support, as well as Virginia Tech's Laboratory for Interdisciplinary Statistical Analysis (LISA) for consulting and help in statistical analysis.

8. REFERENCES

[1] P. Raulefs, "Computational Architectures for Computer-Integrated Engineering and Manufacturing: An Artificial Intelligence Perspective," in *GWAI-89 13th German Workshop on Artificial Intelligence*, D. Metzing, Ed. Springer Berlin Heidelberg, 1989, pp. 455–471.

[2] T. Shibata, "Head mounted display," *Displays*, vol. 23, no. 1, pp. 57–64, 2002.

[3] C. Kiefer, N. Collins, and G. Fitzpatrick, "Evaluating the wiimote as a musical controller," in *Proceedings of the 2008 International Computer Music Conference (ICMC)*, 2008.

[4] "Glass," *Google Developers*. [Online]. Available: https://developers.google.com/glass/. [Accessed: 06-Jan-2017].

[5] M. Silva, D. Freitas, E. Neto, C. Lins, V. Teichrieb, and J. M. Teixeira, "Glassist: using augmented reality on Google Glass as an aid to classroom management," in *Virtual and Augmented Reality (SVR), 2014 XVI Symposium on*, 2014, pp. 37–44.

[6] A. O'Connor, "Google glass enters the operating room," *The New York Times*, 2014.

[7] J. Kastrenakes, "Analog to digital: Cornell conductor tries to modernize the orchestra with Google Glass," *The Verge*, 27-Oct-2013. [Online]. Available: http://www.theverge.com/2013/10/27/5034818/cornell-professor-cynthia-turner-mixes-google-glass-and-classical-music. [Accessed: 03-Jan-2017].

[8] Microsoft, "Microsoft HoloLens," *Microsoft HoloLens*. [Online]. Available: https://www.microsoft.com/microsoft-hololens/en-us. [Accessed: 03-Jan-2017].

[9] "SmartEyeglass SDK," *Sony Developer World*. [Online]. Available: https://developer.sony.com/develop/wearables/smarteye glass-sdk/. [Accessed: 03-Jan-2017].

[10] "Augmented Reality - Home | Meta Company." [Online]. Available: https://www.metavision.com/. [Accessed: 03-Jan-2017].

[11] I. I. Bukvic, J. Wilkes, and A. Gräf, "Latest developments with Pd-L2Ork and its Development Branch Purr-Data," presented at the PdCon 2016, New York, NY, USA, 2016.

[12] "FUDI," *Wikipedia, the free encyclopedia*. 30-Mar-2014.

[13] C. McCormick, K. Muddu, and A. Rousseau, "PdDroidParty-Pure Data patches on Android devices," *Retrieved January*, vol. 21, 2014.

[14] D. Iglesia, "MobMuPlat (iOS application)," *Iglesia Intermedia*, 2013.

[15] "Mira: Touch Controller iPad App For Max | Cycling '74." [Online]. Available: https://cycling74.com/products/mira/. [Accessed: 06-Jan-2017].

[16] C. Roberts, *Control: Software for end-user interface programming and interactive performance*. Citeseer, 2011.

[17] "Style | Glass," *Google Developers*. [Online]. Available: https://developers.google.com/glass/design/style. [Accessed: 06-Jan-2017].

[18] H. Brenton, M. Gillies, D. Ballin, and D. Chatting, "The uncanny valley: does it exist," in *Proceedings of conference of human computer interaction, workshop on human animated character interaction*, 2005.