EAI Endorsed Transactions
on Creative Technologies
Research Article EAI.EU

# From GUI to AVUI: Situating Audiovisual User Interfaces Within Human-Computer Interaction and Related Fields

Nuno N. Correia[1,*], Atau Tanaka[2]

[1]University of Greenwich and ITI/LARSyS
[2]Goldsmiths, University of London

## Abstract

INTRODUCTION: We propose AudioVisual User Interfaces (AVUI), a novel type of UI linking interaction, sound and image. It extends the concept of Graphical User Interface (GUI) by adding interconnected sound and image.
OBJECTIVES: We aim to situate AVUIs in relation to identified relevant fields: Human-Computer Interaction (HCI), sonic interaction design, cognitive psychology and audiovisual art, and to identify benefits of AVUIs in the context of those fields.
METHODS: In this research, we combine literature review of related concepts with a user-centered design methodology, involving a community of audiovisual artists and performers.
RESULTS: We contextualize AVUI within the identified fields and identify benefits for the implementation of AVUIs in relation to those fields.
CONCLUSION: These results can contribute to the further adoption of AVUIs by the HCI community, particularly those interested in multisensory experience involving sound and image.

Received on 19 February 2021; accepted on 08 May 2021; published on 12 May 2021

**Keywords:** user interface, toolkit, interface builder, audiovisual, cross-modal interaction



doi:10.4108/eai.12-5-2021.169913

## 1. Introduction

Audiovisual relationships have been explored in the field of Human-Computer Interaction (HCI) to enhance the user experience and usability, across different application areas – such as accessibility in assistive displays [1], improvement of task accuracy in driving [2], alarms for surveillance activities [3] and enjoyability and performance in games [4, 5]. To facilitate the implementation of congruent audiovisual feedback in interaction design, and its integration with user interfaces, we propose a new type of UI: AVUI (AudioVisual User Interface) [6]. The concept of AVUI links interaction, sound and image, building upon the concept of Graphical User Interface (GUI) by adding interconnected audio and visuals. In research leading up to this article, we have presented the design, deployment and evaluation of the AVUI Toolkit, following a User-Centered Design (UCD) process with audiovisual artists [7]. In this article, we situate AVUIs in relation to relevant scientific fields: HCI, sonic interaction design, cognitive psychology and audiovisual art. We identify benefits of AVUIs and propose AVUIs as a valuable tool for the HCI community.

## 2. Related concepts

We begin by identifying relevant areas to situate and discuss AVUIs within HCI. First, we discuss topics related to HCI, sonic interaction design and cognitive psychology, respectively: GUIs and toolkits; auditory icons and earcons; cross-modal interaction and UI. Next, we approach topics related with audiovisual (AV) art: AV performance; AV tools and Do-It-Yourself (DIY) culture; and finally, AV tools and UI.

*Corresponding author. Email: n.correia@greenwich.ac.uk

## 2.1. GUIs and toolkits

The idea of a "direct manipulation interface", where visible objects on the screen are directly manipulated with a pointing device, was first demonstrated by Ivan Sutherland in Sketchpad in 1963 [8]. In the early 1960s, Douglas Engelbart conceptualised the GUI, using as an example a system with a "pointer". Engelbart would demonstrate his GUI concepts at his famous 1968 demo, "The Mother of All Demos"[1]. Those ideas were further developed at the Xerox Palo Alto Research Center (PARC). Among the many outputs of PARC were the Alto computer and the SmallTalk programming language and development environment, which introduced many modern GUI concepts, such as overlapping windows, icons, popup menus, scroll bars, radio buttons and dialog boxes [9]. These concepts would be influential for the Apple Lisa, Apple Macintosh and Microsoft Windows operating systems, and other GUIs that would follow.

Interface builder toolkits are software development frameworks that are "interactive tools that allow interfaces composed of widgets such as buttons, menus and scrollbars to be placed using a mouse" [10]. Examples of early interface builders are the MenuLay system developed by Bill Buxton at the University of Toronto (1983); the "Resource Editor" included in the original Macintosh (1984), which allowed widgets to be placed and edited; and NeXT Interface Builder (1988) [10]. The Apple Macintosh (1984) was the first operating system to promote its toolkit for use by other developers to ensure a consistent interface [10]. The NeXT Interface Builder was later incorporated into the Apple Xcode environment and is still used for developing interfaces across the different Apple operating systems. With the introduction of the iPhone in 2007 and the iPad in 2010, Apple would popularise a style of interaction for multi-touch screens, which no longer adopts the WIMP (Windows, Icons, Menus, Pointer) model. There are guidelines to implement a UI on software for major operating systems, such as for Apple Mac[2], Apple iOS[3], Android[4] and Windows[5] operating systems.

[1] https://www.dougengelbart.org/content/view/209/448/
[2] https://developer.apple.com/design/human-interface-guidelines/macos/overview/themes/
[3] https://developer.apple.com/design/human-interface-guidelines/ios/overview/themes/
[4] https://developer.android.com/guide/topics/ui
[5] https://docs.microsoft.com/en-gb/windows/win32/windows-application-ui-development

## 2.2. Sonic interaction design, auditory icons and earcons

Sound has a rich history as a medium to enhance user interfaces [11]. The field of sonic interaction design "explores ways in which sound can be used to convey information, meaning and aesthetic and emotional qualities in interactive contexts", where "in order to foster an embodied experience, both the interface and its sonic behaviour must be carefully designed" [12]. One early line of sonic interaction design research proposes "to use sound in a way that is analogous to the use of visual icons to provide information" [13]. The concept of auditory icons aims to "provide a natural way to represent dimensional data as well as conceptual objects in a computer system" [13]. Using an example by Gaver:

> The file hits the mailbox, causing it to emit a characteristic sound. Because it is a large message, it makes a rather weighty sound. The crackle of paper indicates a text file [13]

.

The concept of Auditory Icons was further elaborated, leading to the SonicFinder interface, where information is conveyed using auditory icons as well as visual feedback. Its aims are "an increase in direct engagement with the model world of the computer" and to provide sound information that is consistent with the visual component [11]. In turn, the audio and visualisation would reflect the status of the UI elements, aiming to make "the model world of the computer more real" and "the existence of an interface to that world less noticeable" [11]. Sumikawa coined the term *earcons* as icons for the ear, "audio cues used in the computer-user interface to provide information and feedback to the user about some computer object, operation, or interaction" [14]. Earcons are composed of motives, structured as modules: "single pitches or rhythmicized sequences of pitches" [15]. For example, a sequence of bleeps increasing in pitch when logging in an application and a sequence of bleeps decreasing in pitch when logging out. Therefore, if auditory icons are composed of sounds extracted from (or informed by) the real world, earcons are more abstract.

## 2.3. Cross-modal interaction and multisensory experience

Cross-modal interaction is the phenomenon by which information from one sensory modality influences the processing of signals from another modality [5]. Research on brain plasticity and sensory substitution has explored how the brain replaces functions of one sense by another [16]. There has been a growing interest in the implications of cross-modal interactions

for UI design, particularly involving sound. One study revealed that people's perception of flashing lights can be manipulated by sounds – a single flash of light can be seen as consisting of two flashes if displayed simultaneously with multiple sound signals [17]. How sounds are presented can also influence the number of vibrotactile triggers that a person will perceive [18]. The related issue of congruency – non-arbitrary associations between different modalities – has also been explored in HCI. The perceived quality of touchscreen buttons has been correlated to congruence between visual and audio/tactile feedback used to represent them [19]. In our research with memory games, we found that congruent display across audio and visual modalities showed higher engagement results than arbitrary associations between sound and image [5].

AVUIs relate to multimodal interfaces, which "process two or more combined user input modes" in coordination with a "multimedia system output" [20]. But AVUIs focus on multimodality in the system output rather than on the input modes. Therefore, multisensory user experience – the experience of the multimodal output – is more relevant for our research.

In HCI, the emerging field of multisensory user experience is dedicated to studying experiences "designed with the senses (e.g. sight, hearing, touch, taste, smell) in mind and which can be enabled through existing and emerging technologies" [21]. There is increasing research on multisensory experiences in order to arrive at more generic guidelines and recommendations to design multisensory output in interactive systems, such as Velasco and Obrist's "laws of multisensory experiences" [21].

## 2.4. AV performance

The availability of affordable personal computers capable of real time processing of both audio and video, since the 1990s, gave a further impulse to audiovisual performance [22]. Club culture also has an important role in its development. Here an important distinction should be made, between VJing and audiovisual (AV) performance. VJing (derived from VJ or Video Jockey) has its origins in the club culture of the 1990s, where a visual performer complements the DJ in the club, that is, "because of the absence of a stage act there was a demand for a new visual experience" [23]. VJing has since expanded beyond the club and into other types of music performances. Audiovisual performance implies the combined live manipulation of sound and image [24].

Three notable examples of contemporary AV artists using computer-generated graphics and sound are Golan Levin, Thor Magnusson and Toshio Iwai. They are relevant to this study because they are concerned with creating interfaces and systems for audiovisual performance. Levin developed a group of works under the name *Audiovisual Environment Suite* and described his approach to audiovisual performance as being based on painterly interfaces [25], where the action of drawing generates related sounds. Magnusson uses unconventional GUIs and "abstract objects that move, rotate, blink/bang or interact" to represent musical structures [26]. The tools he develops are often made available online. Iwai creates playful pieces, crossing genres between game, installation, performance (with works such as *Elektroplankton* and *Composition on the Table*) and audiovisual instrument (with *Tenori-On*) [27].

## 2.5. AV tools and DIY culture

Most commercial software tools for VJing, such as Modul8[6] or Resolume[7] focus on video playback and manipulation, with limited generative graphics capabilities, and include only "fairly low-level musical features" [28]. Artists dealing with audiovisual performance, therefore, "often rely on building their own systems using visual programming environments such as Max/MSP/Jitter[8], VVVV[9] or PD[10]" [28]. Others use creative coding frameworks such as Processing[11] or openFrameworks[12]. Therefore, one important element of audiovisual performance is the use of DIY (Do-It-Yourself) tools [22].

Processing and openFrameworks are important examples of tools aimed specifically to empower artists and designers to develop their own software. The *Design by Numbers* project, initiated in the 1990s by John Maeda and his students at the MIT Media Lab, was aimed to empower "visual people", designers and artists, to code [29]. Since 2001, two of his students, Casey Reas and Ben Fry, further pursued these aims with a new project, Processing, which has become a popular development tool for media arts [30]. Processing inspired other technologies, such as openFrameworks, based on the C++ programming language. In addition to the objectives of Processing, openFrameworks aimed to provide more low-level access to computational devices for more demanding tasks and take advantage of numerous software libraries written in the C++ language [31].

The emergence of the World Wide Web in the 1990s increased the dissemination of DIY tools, and now artists are modifying them, "creating tools within tools,

[6] http://www.garagecube.com/modul8/
[7] https://resolume.com
[8] https://cycling74.com
[9] https://vvvv.org
[10] https://puredata.info
[11] https://processing.org
[12] https://openframeworks.cc

and sharing these developments with others" [32]. Despite the potential of these creative coding environments, "this approach requires substantial scratch-building and a high level of technical ability on the part of the user" [28]. There is therefore a need for software bridging between ready-made commercial software and DIY solutions using programming environments.

### 2.6. AV tools and UI

Following the tradition of interface builders, and to facilitate the implementation and reusability of GUIs, UI toolkits have been developed for coding platforms such as Processing and openFrameworks. These external software toolkits or libraries expand the functionality of these platforms and need to be imported and linked to in the code before they are used. These are entitled "libraries" in Processing and "addons" in openFrameworks (by convention, the latter have the prefix "ofx"). Popular UI libraries include ControlP5[13] for Processing and ofxGUI, distributed by default with openFrameworks. UI toolkits have also been developed for the web, often using JavaScript, such as Interface.js [33], a cross-platform library for touch, mouse and motion events oriented toward live performance.

Traditionally, media content and GUI have been displayed separately – the creative output of any media production tool (the "content") is distinct and separate from the control mechanisms (the "user interface") that produce it. Image processing software such as Photoshop or Modul8, for example, rely on windows or "tool palettes" to organize the GUI, while the graphical content is shown in its own window or dedicated area. Some tools, such as off-the-shelf VJ software, rely on a dual-screen logic: the user or performer sees the GUI and a preview of the video output in one screen, while the final video output is displayed in full on a second screen, without any GUI.

With the emergence of touch screen computational devices such as tablets, another type of software appeared: "controller" software for remote manipulation of another software typically running on a laptop. These controller applications usually consist of GUI builders, a back end to rearrange and map the GUI to features on the remote software, and communication settings to establish a connection with the computer (typically using the OSC protocol). Examples of these tablet software controllers: TouchOSC[14], Lemur[15] and touchAble[16]. However, with functional aesthetics subscribing to the adage, "form follows function", we began to explore the possibility of the interface being itself embedded in the creative output of a work, or the aesthetic content reflecting the interaction dynamics that bear it.

## 3. AVUIs

The *Enabling Audiovisual User Interfaces for Multisensorial Interaction* project took place between 2014 and 2016. During the project, we developed AVUIs not only as a toolkit but also as a set of design guidelines for practitioners wishing to make creative audiovisual work where interface and content were fused [6]. This development involved the participation of artists and developers in a User-Centered Design process. The hypothesis behind the research was: "the introduction of AVUI, integrating interrelated sonic and visual feedback, reacting to user interactions, will lead to more usable, accessible, playful and engaging UIs, as compared to a traditional GUI – particularly in use cases where accessibility and/or engagement are determinant"[17]. In a process with multiple stages (such as interviews, workshops [7], hackathons [34] and performances [35]), we involved a community of audiovisual performers, since they are experts in combining sound, image and interactivity. This work expands upon the notion of Interactive Audiovisual Objects: "integration of sound, audio visualization and graphical user interface into modular units" [36], developed by the first author for his audiovisual performance practice. But AVUIs aim to allow for a more flexible UI, building upon the tradition of UI toolkits, and more general applicability.

We used openFrameworks for the development of the toolkit. We also used the Maximilian addon[18], to extend the audio capabilities of openFrameworks. We divided the code into three groups of class files: audio, visuals and UI. Each of these groups has a base class, facilitating to extend and create new audio processes, new visualizations and new UI types. It was released as an addon, allowing to be easily integrated in openFrameworks projects by developers. We released the addon in versions for personal computers and mobile devices. The ofxAVUI addon was released as open source in our GitHub repository[19]. It is now also part of the main directory for openFrameworks addons, in the UI category[20]. As is usual with openFrameworks addons, we included examples and extensively commented the source code.

We adopted the notion of "zones" as an organization structure for combining sound, image and UI. Each

[13] http://www.sojamo.de/libraries/controlP5/
[14] http://hexler.net/software/touchosc
[15] https://liine.net/en/products/lemur
[16] https://zerodebug.com/
[17] http://avuis.goldsmithsdigital.com
[18] https://github.com/micknoise/Maximilian
[19] https://github.com/AVUIs/ofxAVUI
[20] http://ofxaddons.com/categories/1-gui

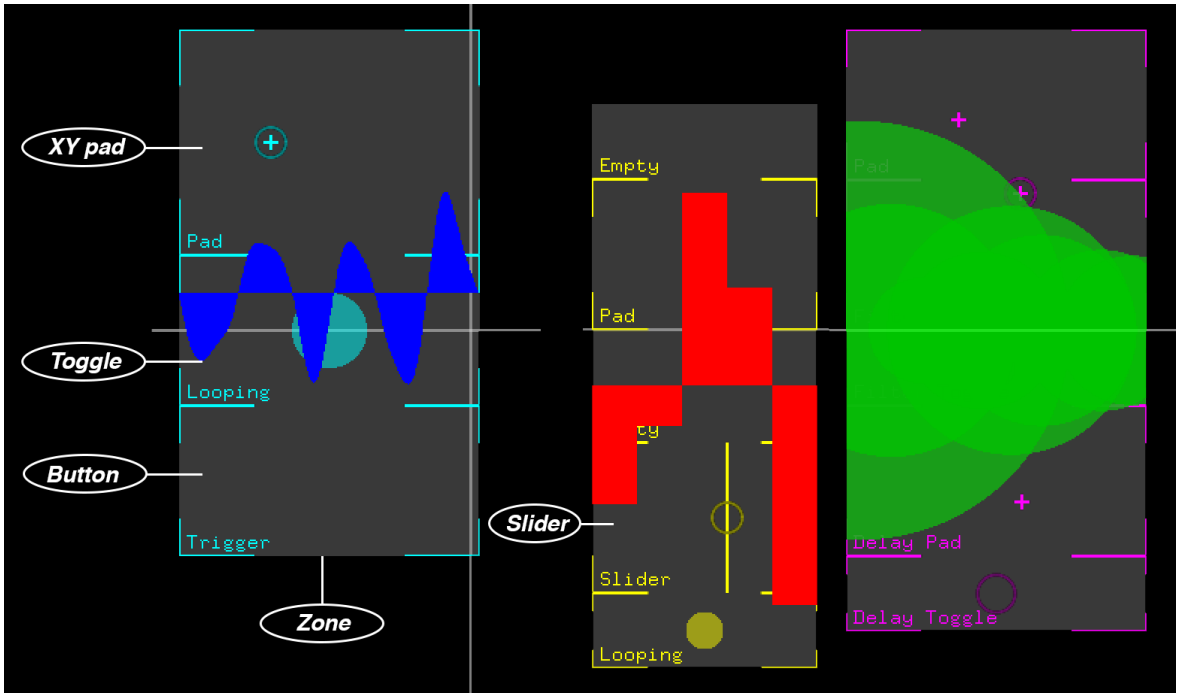


**Figure 1.** Example of an AVUI layout, with captions pointing to the different UI elements

zone has only one sound and one visualization, to reinforce its individuality and its objecthood. Different UI elements can be added to a zone: buttons, toggles, XY pads, sliders, range sliders, drop-down menus and labels. The number of zones can be specified, as well as their size, position, color palette and UI elements. The example shown in Figure 1 showcases three zones, each with different UI elements (as shown in the captions), different sizes, positions, visualizations (waveform, bars and circles) and color palettes.

The example shown in Figure 2 shows the implementation of AVUI in the audiovisual performance app *AV Zones* (for iOS). There are 3 zones, each with the same UI structure and same visualization (waveform), but with different sounds and colors. The respective UI is composed of labels (at the top), toggles (shorter elements), XY pads (taller elements) and a read head (vertical red line). Additional circles represent points touched by the user.

We included three visualizations with the addon, with further configuration options. These default visualizations rely on a direct mapping of the audio buffer data into two-dimensional graphical data for drawing basic vectors – lines, rectangles and circles (representing amplitude of sound over time). We also facilitated the creation of new visualizations, making the visualization module extensible. For example, FFT and MFCC visualizations have been created, allowing to map sound frequency data to graphics.

Any parameter from the UI can be redirected to any audio feature of the zone, or any other aspect of the software (for example, any graphic on the screen). These design elements, essential to the definition of an AVUI, are exposed to the openFrameworks developer through high level function calls, making integration into an openFrameworks project straightforward.

As mentioned, the AVUIs toolkit is available on GitHub. Is has been featured on the ofx addons gallery[21] and has been used by 32 developers (estimate based on GitHub analytics). During our development, six projects were created with AVUI by test users: *FFT/MFCC*, audio frequency analyzers and visualizers; *Step Sequencer* for creating rhythmic patterns; *Background Image*, for customizing the appearance of zones; *Lisajous and Grid*, two additional visualizers; a four-zone *Multisampler*; and *ShaderUI*, an implementation of sound-responsive shaders. It has been used to built an AV instrument for iOS by the first author, *AV Zones*, presented in a series of performances, and was the core library with which the iOS app *ShapeTones* was developed and published on the Apple App Store.

## 4. Discussion

We have situated the concept of AVUI as being in the intersection of the identified related fields, notably HCI (Figure 3). We will next discuss AVUI and compare this UI type to these related concepts.

[21] https://ofxaddons.com

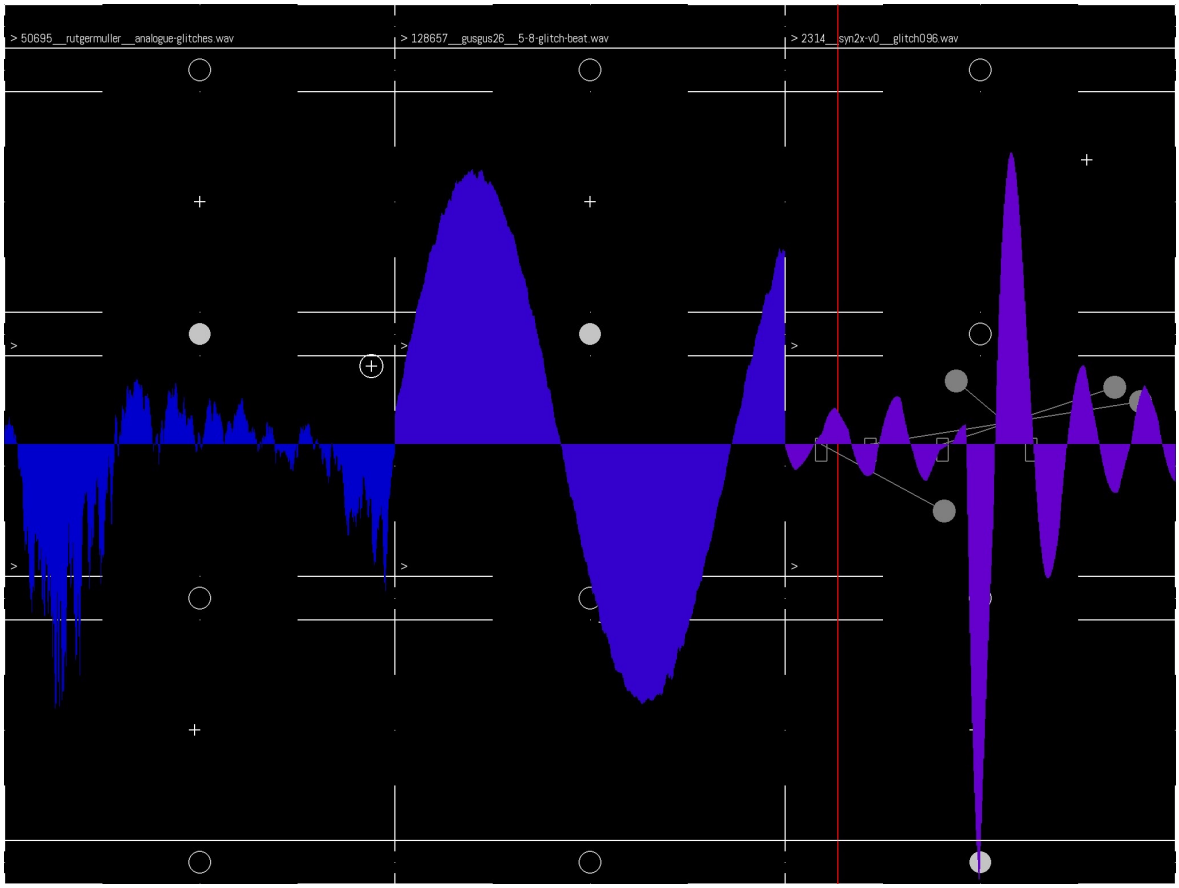


**Figure 2.** Implementation of AVUI in the audiovisual performance app *AV Zones*

### 4.1. Leveraging cross-modal interaction to combine sound with GUI

With AVUIs, it is possible to combine the sonic approach of auditory icons [13] and earcons [14] with visual counterparts, in an interrelated and congruent way. Therefore, AVUIs allow to create "audiovisual icons" as well as other user interfaces elements, leveraging cross-modal interaction [5] and sensory redundancy to reinforce their feedback across visual and auditory senses. The affordances of AVUIs to map parameters between UI, sound and image (and any other aspect of the application environment) facilitate conveying dimensional data, and interactions with it, through both sound and image. For example, an AVUI scrollbar could, when scrolling through folders that contain larger files, contain a visualization that would go darker in color and an associated sonification that would go lower in pitch (and vice-versa when scrolling through folders containing smaller files).

### 4.2. From AV performance to interface design...

As seen in the Introduction section, the application areas for this type of UI go potentially beyond art, into fields such as games [5] and accessibility [1]. However, our approach with AVUI development was to co-design with audiovisual artists. Our premise was that, by designing AVUIs with artists, we would tap into their expertise on combining sound, image and real-time interaction to create more generic audiovisual interfaces. Audiovisual artists are experts in expression across modalities and have tacit knowledge of the cross-modal effects, which we aim to leverage. Wanderley et al. state: "Expert musicians push the boundaries of system design through their personalization and appropriation of music applications" [37]. Similarly, we believe that audiovisual artists also push the boundaries of system design, and that the knowledge gained from analyzing this can be useful for more generic interfaces. The assumption is that, if a system is robust enough for the high demand of performers, it will have passed an important test toward more generic application.

### 4.3. ...and from interface design back to AV performance

AVUIs are also relevant for creating systems for audiovisual performance. As Malloch et al. state, "the creative context of music provides opportunities for putting cutting-edge HCI models and tools into

practice" [38], and we argue that the same is true for the context of audiovisual performance. Our AVUI toolkit uses openFrameworks, a programming environment that is widely adopted by audiovisual artists to create their own systems and tools – therefore facilitating adoption by this community. We organized a hackathon with eight audiovisual performers to assess the potential of ofxAVUI to create their own systems with the toolkit [6]. We also contacted the online community of ofxAVUI users on GitHub and two artists agreed to test it by creating their own systems. The outcomes of these different tests demonstrate that the toolkit facilitates speed and ease of development of audiovisual performance systems, among other identified benefits [6]. It also allows to communicate the interface of the software, and therefore the agency of the performer, to the audience of audiovisual performances. This approach has been successful in two audience studies we have conducted [35]. We created our own proof of concept with ofxAVUI for audiovisual performance, an application for Apple iOS entitled *AV Zones*, and the first author has performed extensively with it [39].

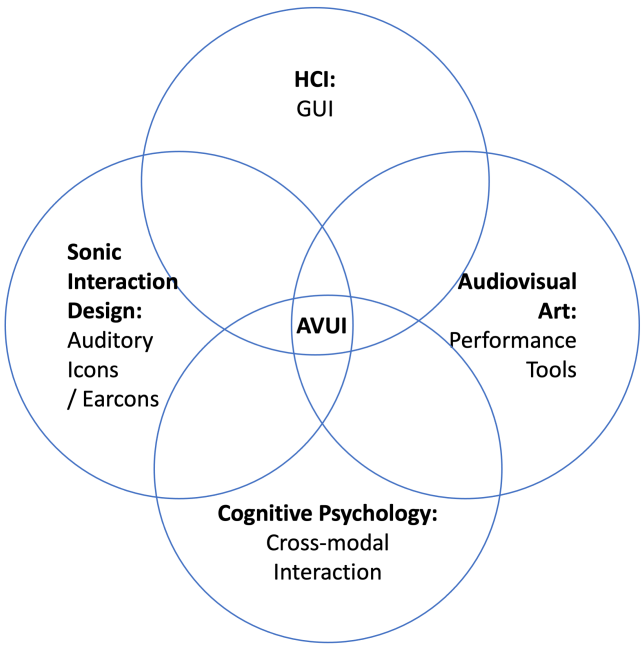


**Figure 3.** Diagram situating AVUI in the intersection of the identified related fields

## 5. Conclusion

We have presented the concept of AudioVisual User Interface (AVUI), we identified fields related to it – namely: HCI, sonic interaction design, audiovisual art and cognitive psychology – and we discussed AVUIs in relation to each of these fields. The main contribution and novelty of this paper is situating AVUI at the intersection of those fields, allowing us to identify benefits of AVUIs compared to related approaches:

- AVUIs allow to combine "icon-based" sonic interaction design approaches (such as auditory icons and earcons) with congruent visualization, and extend it to other UI elements.
- AVUIs have demonstrated reliability and flexibility, as they have been developed with artists who are experts in combining audio and image, with associated high demands in terms of usability.
- AVUIs can assist in terms of speed and ease of development of systems for creative audiovisual scenarios.

AVUIs allow for a perception of direct manipulation of sound and image, leading to a more expressive interaction, "a sensation akin to being in direct contact or touching and molding media" [40]. AVUIs have the potential to represent dimensional data in both visual and sonic domains, building upon the premise of auditory icons [13], with the added reinforcement of a dual sensory representation. They can also reinforce the pitch trajectories common in earcons [15] - for example, a sequence rising in pitch (common for activating a system) can be accompanied by a rising visualization. AVUIs also can be an important tool for designing multisensory user experiences, an area which is growing in interest within HCI [21].

The AVUI concept is creatively driven by a functional aesthetic that explores the convergence of the interface with artistic output. This proposition is not limited to aesthetics, but is useful in interface design – be it in the pursuit of elegance, or in fulfilling the need to visualize interaction to the end user. The AVUI concept was embodied in an interface builder type software toolkit, ofxAVUI, that shows its applicability in each of these domains. This toolkit has been used in interaction research, adopted by audiovisual artists including the first author, and was a core library in a commercial mobile app. The library is available to designers, artists, and developers wishing to adopt the design principles presented here. We hope that this will contribute to the further adoption of AVUIs by the HCI community, particularly those interested in multisensory experience involving sound and image.

Regarding future work, we would like to implement AVUIs in other creative coding environments, such as Processing and PureData, as well as to develop a more generic JavaScript library, integrated with the Web Audio API [22]. We also would like to extend the

[22] https://developer.mozilla.org/en-US/docs/Web/API/Web_Audio_API

research by developing other case studies involving AVUIs, for example related to information sonification and visualization. We would also like to improve the usability of AVUI development by creating a graphical editor where UIs could be more easily patched and mapped to image and sound properties.

**Acknowledgement.** This work was supported by the EU Marie Curie fellowship FP7 REA grant 627922 and hosted at Goldsmiths, University of London. This research was also funded by LARSyS, project UIDB/50009/2020.

## References

[1] Ho-Ching, F.W.L., Mankoff, J. and Landay, J.A. (2003) Can you see what i hear?: the design and evaluation of a peripheral sound display for the deaf. In *Proc. CHI'03*, CHI '03 (New York, NY, USA: ACM): 161–168. doi:10.1145/642611.642641, URL http://doi.acm.org/10.1145/642611.642641.

[2] Onimaru, S., Uraoka, T., Matsuzaki, N. and Kitazaki, M. (2008) Cross-modal information display to improve driving performance. In *Proc. ACM symposium on Virtual reality software and technology*, VRST '08 (New York, NY, USA: ACM): 281–282. doi:10.1145/1450579.1450653, URL http://doi.acm.org/10.1145/1450579.1450653.

[3] Audry, E. and Garcia, J. (2019) Congruent audio-visual alarms for supervision tasks. In *Proc. ICAD 19, 25th international conference on auditory display*. URL https://hal-enac.archives-ouvertes.fr/hal-02135341.

[4] Collins, K. and Taillon, P.J. (2012) Visualized sound effect icons for improved multimedia accessibility: A pilot study. *Entertainment Computing* **3**(1): 11–17. doi:10.1016/j.entcom.2011.09.002, URL http://www.sciencedirect.com/science/article/pii/S1875952111000309.

[5] Metatla, O., Correia, N.N., Bryan-Kinns, N., Stockman, T. and Martin, F. (2016) Tap the ShapeTones: Exploring the effects of crossmodal congruence in an audio-visual interface. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '16 (San Jose, California, USA: ACM). URL http://dl.acm.org/citation.cfm?id=2858456.

[6] Correia, N.N. and Tanaka, A. (2017) AVUI: Designing a Toolkit for Audiovisual Interfaces. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17 (Denver, CO, USA: ACM). URL https://dl.acm.org/doi/abs/10.1145/3025453.3026042.

[7] Correia, N.N. and Tanaka, A. (2014) User-Centered Design of a Tool for Interactive Computer-Generated Audiovisuals. In *Proc. 2nd ICLI* (Lisbon: ICLI): 1–14. URL http://www.liveinterfaces.org/proceedings2014.html.

[8] Sutherland, I.E. (1963) *Sketchpad, a man-machine graphical communication system*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA.

[9] Reimer, J. (2005), A History of the GUI. URL http://arstechnica.com/features/2005/05/gui/.

[10] Myers, B.A. (1996) User interface software technology. *ACM Computing Surveys* **28**(1): 189–191. doi:10.1145/234313.234394, URL https://doi.org/10.1145/234313.234394.

[11] Gaver, W.W. (1989) The SonicFinder: An Interface That Uses Auditory Icons. *Hum.-Comput. Interact.* **4**(1): 67–94. doi:10.1207/s15327051hci0401_3, URL http://dx.doi.org/10.1207/s15327051hci0401_3.

[12] Franinovic, K. and Salter, C. (2013) The Experience of Sonic Interaction. In Franinovic, K. and Serafin, S. [eds.] *Sonic Interaction Design* (Cambridge, MA: The MIT Press), 39–76.

[13] Gaver, W.W. (1986) Auditory Icons: Using Sound in Computer Interfaces. *Hum.-Comput. Interact.* **2**(2): 167–177. doi:10.1207/s15327051hci0202_3, URL http://dx.doi.org/10.1207/s15327051hci0202_3.

[14] Sumikawa, D.A. (1985) *Guidelines for the integration of audio cues into computer user interfaces*. Tech. Rep. UCRL-53656, Lawrence Livermore National Lab., CA (USA). URL https://www.osti.gov/biblio/5475406-guidelines-integration-audio-cues-computer-user-

[15] Blattner, M.M., Sumikawa, D.A. and Greenberg, R.M. (1989) Earcons and Icons: Their Structure and Common Design Principles. *Human–Computer Interaction* **4**(1): 11–44. doi:10.1207/s15327051hci0401_1, URL https://www.tandfonline.com/doi/abs/10.1207/s15327051hci0401_1.

[16] Bach-y Rita, P. (1987) Brain Plasticity as a Basis of Sensory Substitution. *Neurorehabilitation and Neural Repair* **1**(2): 67–71. doi:10.1177/136140968700100202, URL http://nnr.sagepub.com/content/1/2/67.

[17] Shams, L., Kamitani, Y. and Shimojo, S. (2000) Illusions: What you see is what you hear. *Nature* **408**(6814): 788–788. doi:10.1038/35048669, URL http://www.nature.com/nature/journal/v408/n6814/abs/408788a0.html.

[18] Lee, J.H. and Spence, C. (2009) Feeling what you hear: task-irrelevant sounds modulate tactile perception delivered via a touch screen. *Journal on Multimodal User Interfaces* **2**(3-4): 145. doi:10.1007/s12193-009-0014-8, URL http://link.springer.com/article/10.1007/s12193-009-0014-8.

[19] Hoggan, E., Kaaresoja, T., Laitinen, P. and Brewster, S. (2008) Crossmodal Congruence: The Look, Feel and Sound of Touchscreen Widgets. In *Proceedings of the 10th International Conference on Multimodal Interfaces*, ICMI '08 (New York, NY, USA: ACM): 157–164. doi:10.1145/1452392.1452423, URL http://doi.acm.org/10.1145/1452392.1452423.

[20] Oviatt, S. (2003) Multimodal interfaces. In Jacko, J.A. and Sears, A. [eds.] *The human-computer interaction handbook* (Hillsdale, NJ, USA: L. Erlbaum Associates Inc.), 286–304. URL http://dl.acm.org/citation.cfm?id=772072.772093.

[21] Velasco, C. and Obrist, M. (2020) *Multisensory Experiences: Where the senses meet technology* (New York, NY: OUP Oxford).

[22] Salter, C. (2010) *Entangled: Technology and the Transformation of Performance* (Massachusetts: MIT Press).

[23] Crevits, B. (2006) The Roots of VJing. In Faulkner, M. [ed.] *VJ: Audio-Visual Art and VJ Culture* (London:

Laurence King Publishing), 14–19.

[24] Carvalho, A. and Lund, C. [eds.] (2015) *The Audiovisual Breakthrough* (Berlin: Fluctuating Images). URL http://www.ephemeral-expanded.net/audiovisualbreakthrough/.

[25] Levin, G. (2000) *Painterly Interfaces for Audiovisual Performance*. Ph.D. thesis, M.Sc. Dissertation. Massachusetts Institute of Technology, Cambridge, MA. URL http://acg.media.mit.edu/people/golan/thesis/thesis300.pdf.

[26] Magnusson, T. (2006) Screen-Based Musical Instruments as Semiotic Machines. In *Proc. of NIME '06* (Paris). URL http://www.ixi-software.net/thor/semioticmachines.pdf.

[27] Nagle, P. (2008) Yamaha Tenori-On Review. *Sound On Sound* (Feb. 2008). URL http://www.soundonsound.com/sos/feb08/articles/yamahatenorion.htm.

[28] Schofield, G., Green, D., Smith, T., Wright, P. and Olivier, P. (2014) Cinejack: Using Live Music to Control Narrative Visuals. In *Proceedings of the 2014 Conference on Designing Interactive Systems*, DIS '14 (New York, NY, USA: ACM): 209–218. doi:10.1145/2598510.2598520, URL http://doi.acm.org/10.1145/2598510.2598520.

[29] Maeda, J. (2001) *Design by Numbers* (Cambridge, MA: MIT Press). URL https://mitpress.mit.edu/books/design-numbers.

[30] Reas, C. and Fry, B. (2007) *Processing* (Cambridge, MA: MIT Press). URL https://mitpress.mit.edu/books/processing.

[31] Lieberman, Z., ofBook - Foreword. URL http://openframeworks.cc/ofBook/chapters/foreword.html.

[32] deLahunta, S. (2005) Isadora 'almost out of beta': tracing the development of a new software tool for performing artists. *International Journal of Performance Arts and Digital Media* **1**(1): 31–46. doi:10.1386/padm.1.1.31/1, URL http://www.tandfonline.com/doi/abs/10.1386/padm.1.1.31/1.

[33] Roberts, C., Wakefield, G. and Wright, M. (2013) The Web Browser As Synthesizer And Interface. In *Proc. NIME 2013* (Daejeon, Seoul: NIME). URL http://nime2013.kaist.ac.kr/program/papers/day2/paper6/282/282_Paper.pdf.

[34] Correia, N.N. and Tanaka, A. (2015) Prototyping Audiovisual Performance Tools: A Hackathon Approach. In *Proc. NIME 2015* (Baton Rouge, Louisiana). URL http://www.nime.org/proceedings/2015/nime2015_230.pdf.

[35] Correia, N.N., Castro, D. and Tanaka, A. (2017) The Role of Live Visuals in Audience Understanding of Electronic Music Performances. In *Audio Mostly* (New York, NY, USA: ACM Press): 1–8. doi:10.1145/3123514.3123555, URL https://dl.acm.org/citation.cfm?id=3123555.

[36] Correia, N.N. (2013) *Interactive Audiovisual Objects*. Doctoral thesis, Aalto ARTS Books, Helsinki. URL https://aaltodoc.aalto.fi/handle/123456789/11341.

[37] Wanderley, M., Malloch, J., Garcia, J., Wendy, M., Michel, B.L. and Stéphane, H. (2016) Human Computer Interaction meets Computer Music : The MIDWAY Project. In *Music and HCI Workshop, CHI'16 Extended Abstracts on Human Factors in Computing Systems*. (ACM). URL http://research.gold.ac.uk/id/eprint/17304/.

[38] Malloch, J., Garcia, J., Wanderley, M.M., Mackay, W.E., Beaudouin-Lafon, M. and Huot, S. (2019) A Design Workbench for Interactive Music Systems. In Holland, S., Mudd, T., Wilkie-McKenna, K., McPherson, A. and Wanderley, M.M. [eds.] *New Directions in Music and Human-Computer Interaction*, Springer Series on Cultural Computing (Cham: Springer International Publishing), 23–40. doi:10.1007/978-3-319-92069-6_2, URL https://doi.org/10.1007/978-3-319-92069-6_2.

[39] Correia, N.N. (2016) AV Zones – Tablet App for Audiovisual Performance. In *Proc. 3rd ICLI* (Brighton).

[40] Hook, J., Green, D., McCarthy, J., Taylor, S., Wright, P. and Olivier, P. (2011) A VJ centered exploration of expressive interaction. In *Proc. CHI'11*, CHI '11 (New York, NY, USA: ACM): 1265–1274. doi:10.1145/1978942.1979130, URL http://doi.acm.org/10.1145/1978942.1979130.