AVUI: Designing a Toolkit for Audiovisual Interfaces

Nuno N. Correia
Madeira-ITI, U. Madeira
Funchal, Portugal
nuno.correia@staff.uma.pt

Atau Tanaka
Goldsmiths, University of London
London, UK
a.tanaka@gold.ac.uk

ABSTRACT
The combined use of sound and image has a rich history, from audiovisual artworks to research exploring the potential of data visualization and sonification. However, we lack standard tools or guidelines for audiovisual (AV) interaction design, particularly for live performance. We propose the AVUI (AudioVisual User Interface), where sound and image are used together in a cohesive way in the interface; and an enabling technology, the ofxAVUI toolkit. AVUI guidelines and ofxAVUI were developed in a three-stage process, together with AV producers: 1) participatory design activities; 2) prototype development; 3) encapsulation of prototype as a plug-in, evaluation, and roll out. Best practices identified include: reconfigurable interfaces and mappings; object-oriented packaging of AV and UI; diverse sound visualization; flexible media manipulation and management. The toolkit and a mobile app developed using it have been released as open-source. Guidelines and toolkit demonstrate the potential of AVUI and offer designers a convenient framework for AV interaction design.

ACM Classification Keywords
H.5.2 User Interfaces: Auditory (non-speech) feedback; H.5.2 User Interfaces: Graphical user interfaces (GUI); H.5.2 User Interfaces: Prototyping; H.5.2 User Interfaces: User-centered design

Author Keywords
User interface; toolkit; interface builder; audiovisual; crossmodal interaction; prototyping; participatory design; interaction design; hackathons.

INTRODUCTION
The combination of audio with image has a long tradition, from color organs used by the composer Scriabin in the early 20th century [27] to the pioneering computer graphics explorations of John Whitney in the 1960s [26]. The advent of powerful personal computers for media manipulation, from the 1990s, gave further impulse to audiovisual (AV) performance [32].

CHI 2017, May 06 – 11, 2017, Denver, CO, USA.

ACM 978-1-4503-4655-9/17/05...$15.00
DOI: http://dx.doi.org/10.1145/3025453.3026042

These artistic explorations mirror how the brain deals with multi-sensorial information. Research on sensory substitution has explored how the brain replaces functions of one sense by another [1]. The perception of flashing lights can be manipulated by sound: a single flash of light can be seen as consisting of two flashes if displayed simultaneously with multiple sound signals [34]. In the McGurk effect, the perception of an auditory phoneme changes depending on the image [24]. An important factor for crossmodality is congruency – non-arbitrary associations between different modalities. A congruent AV display can result in better performance and higher engagement than arbitrary associations between sound and image [25].

Sound and image have been studied in HCI in different application areas, including accessibility in assistive displays [11], improvement of task accuracy in driving [31], enjoyability and performance in games [5, 25]. Despite the potential for facilitating usability and engagement, there is a lack of design guidelines and standard tools for AV interaction design. Specifically, current solutions for AV performance that facilitate UI integration with content are laborious to implement, and lack aesthetic concerns regarding coherence of UI and visuals. Interface design for AV performance is mostly subjective, and there are no established best practices. These best practices would benefit performers, audience, software developers, interaction designers, researchers and students.

We propose the AVUI (AudioVisual User Interface) where the interaction of sound and image in the interface extends the concept of GUIs. We seek to 1) leverage practices in AV performance for sketches and prototypes, using participatory design methods; 2) implement best practices into a consolidated prototype; 3) propose guidelines and a software toolkit to allow designers to easily integrate sound and image in the UI of future systems and products.

This paper reports on the multi-stage design, development, release and evaluation of a software toolkit, ofxAVUI. We present related work; the participatory design and qualitative methods used; and their results; followed by AVUI guidelines proposed, discussion and conclusions.

RELATED WORK

AV Performance and Tools
AV performance combines live manipulation of sound and image [4]. This distinguishes it from VJing (Video Jockey performance), where a visual performer accompanies a musician [8]. A number of artists are concerned with creating interfaces and systems for AV performance. Levin developed

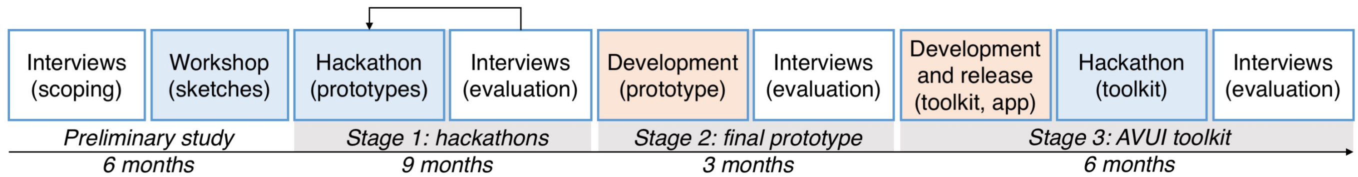

Figure 1. Summary of the four cycles of the research: preliminary study; stages 1, 2 and 3. Workshops and hackathons in blue, interviews in white and development phases in orange. Stage 1 had two iterations

painterly interfaces for audiovisual performance [20]. Magnusson uses abstract GUIs to represent musical structures [23]. Iwai creates playful pieces, crossing genres between game, installation and performance [29].

Most commercial VJ software, such as Modul8 (http://www.modul8.ch), focus on video playback and manipulation, with limited generative graphics capabilities, and only "fairly low-level musical features" [33]. AV performers "often rely on building their own systems" [33] with coding frameworks such as openFrameworks (OF) (http://openframeworks.cc). Therefore, an important element of VJing and AV performance is the use of Do It Yourself (DIY) tools [32]. This requires "a high level of technical ability on the part of the user" [33].

Solutions such as Processing (http://processing.org) offer sound toolkits, which contain visualization modules (ex: Minim), and also offer GUI modules (ex: Control P5). Similar examples could be given for other creative coding environments. But although these UI and sound/visualization modules can be combined, this is laborious, and they are not integrated out of the box, neither functionally nor aesthetically. They do not offer easy ways to implement an "AV+UI" solution, nor a GUI designed to be integrated with visuals. We aim to provide an ease of implementation, and high level of integration.

AV Systems and Interaction Design

Schofield et al. created Cinejack for "directing narrative video through live musical performance" [33], in collaboration with artists. In MelodicBrush, a user-centered design approach is adopted to design a tabletop AV system linking calligraphy and music [15]. Wiethoff and Gehring created an interactive media façade system through an iterative approach: key data collection; user research; data analysis; design concepts; and experience concepts [35]. These studies have used multi-stage, user-centered approaches, which we adopted in our work. However, they each only evaluated one system, making generalization difficult. We apply these methods to evaluate a large number of projects in order to glean design insights across multiple systems.

The New Interfaces for Musical Expression (NIME) community has been active in studying the combination of visuals, sound and interaction design. The work of Jordà, such as FMOL and Reactable, is relevant due to the interplay of interaction and sound visualization strategies [17]. The authors of residUUm, one of the prototypes resulting from the current research, have presented their performance approaches for AV generation and audience visibility [30]. Rouages is another AV project that is concerned with audience understanding [2].

Hook has developed an interface for VJ performances, Waves [14]. He also studied VJing from the viewpoint of the performer [13] and identified three main themes: *aspirational*, *live* and *interaction*. Within *interaction*, he identifies the following sub-themes: *constraining interactions*; *haptically direct*; *parallel interaction*; *immediacy*; *manipulable media*; *reconfigurable interfaces*; and *visible interaction*. While he focuses on video content, he recognizes the need for generative media tools. We will use Hook's framework for our study, strengthening the potential generalizability of the work by building upon prior qualitative findings.

METHODS

The research took place over a 2-year period and involved a preliminary study and three main stages (Figure 1): Pre) scoping interviews and brainstorming workshop; 1) hackathons for prototype development; 2) a "consolidation prototype" gathering best practices and expert evaluation; 3) creation of a software development toolkit for facilitating integration of AV with UI. The toolkit was released as open-source, tested in a hackathon and with internet users, and further tested by rebuilding the final prototype from Stage 2 as a publicly released product using the toolkit. The iterative cycle enabled "a dynamic process of invention, distributed across events" [9]. Hackathons and interviews (and subsequent thematic analysis) were used as methods throughout the studies.

Hackathons

Hackathons are coding events in DIY communities where "small teams produce working software prototypes in a short time period" and these events are often centered around a common theme or technology [18]. The hackathon challenge is an important part of the method. It sets a common task in a motivating way to participants, making a hackathon "a moment of design" [16]. In a hackathon, solutions are "conceived in response to those challenges" [21]. These elements make it a fun, easy to understand technique for participatory design and code development.

Thematic Analysis

We conducted thematic analyses of interviews in the three studies, based on techniques in [3]. We coded the responses based on emerging patterns and issues arising, then collated the codes into potential themes. We used Hook's themes as a basis for our coding. We complemented this theoretic analysis approach with an inductive analysis independent of any pre-existing coding frame. This allowed us to build on prior work, contribute our new insights, and achieve a balanced, thorough and inclusive structure of main themes and sub-themes.

PRELIMINARY STUDY: INTERVIEWS, BRAINSTORMING

In our preliminary study [7], we conducted interviews with 12 audiovisual performers, asking them about their practice, their tools, and their needs and desires as performers. The analysis of the interviews brought forth a series of key issues: modularity, flexibility and reconfigurability; ease of hardware/software integration; instrument-like expressivity and fluidity; integration of environmental elements; generative capabilities and diversity; communication of process to the audience; reliability and speed. These concepts on the whole match and confirm the issues identified by Hook under the theme *interaction*. Generative capabilities and diversity connects to their forward looking theme of *aspiration*, and the need for a visual equivalent to sound synthesizers. The 12 interviewees provided us with a group of experts that we would consult throughout the different studies of the research – the evaluators of Stages 1 and 2 were from this same group.

The ideas from the interviews then informed a brainstorming workshop, with 19 participants (including two from the previous interview stage). The one-day workshop structure was comprised of two parts: the first one adopting the "bootlegging" idea generation technique [12]. For part 2, we extended this with a more focused, structured re-examination of ideas from part 1, which we called "Re-boot".

The five breakout groups produced five sketches (storyboards and wireframes) of procedural audiovisual performance tools. Two sketches, *Gestural Touchscreen* and *Meta/Vis*, were particularly successful in addressing the challenges set out in the workshop. Both rely on the expressive potential of multitouch interaction, employing different solutions for reconfigurability: the former allows for loading and manipulating vector graphics, and the latter adopts a simplified data-flow mechanism. Project descriptions and sketches are seen at `http://www.gen-av.org/sketches/`.

STAGE 1: PARTICIPATORY DESIGN OF PROTOTYPES

Hackathon and Hack Challenge

Using the key themes and the sketches from the preliminary study as input and inspiration, we ran two hackathons in an iterative cycle, Gen.AV 1 and Gen.AV 2. The objective was to leverage knowledge from AV performers into prototypes combining AV and UI, where best practices could be adopted in a future AVUI toolkit. We sent out a call for participation, with coding knowledge as prerequisite, and interviewed applicants. Each hackathon took place over two days.

Both Gen.AV 1 and 2 followed the same structure: 1) Introduction: a presentation on the previous stages of the study and results achieved so far, goals and structure of the workshop; 2) Conceptualization and sketching; and 3) Software development. 23 participants took part in Gen.AV 1 (five female and 18 male). Gen.AV 2 had 13 participants (two female and 11 male), three of whom had taken part in Gen.AV 1. We divided participants into five (Gen.AV 1) and six (Gen.AV 2) groups, distributing prior programming experience evenly across groups.

We created hack challenges based on key conclusions (*in parenthesis*) from the preliminary study [7]. They were: 1) computer-generated sound and visuals (*generative capabilities and diversity*); 2) powerful and fluid manipulation – "like an instrument" (*instrument-like expressivity and fluidity*); 3) single-screen – what the performer sees is what the audience sees (*communication of process to the audience*); and 4) possibility to reconfigure the interface (*modularity, flexibility and reconfigurability*). The resulting projects were presented in two public performances. Five projects were showcased in the Gen.AV 1 performance, and six in Gen.AV 2. Each group produced a 10 minute performance.

Projects

We present the five projects from Gen.AV 1. *ABP* is an animation engine and sound visualizer, where the user can define color, geometry and animation parameters. In *drawSynth*, a GUI allows controlling sound and image – users can draw vector shapes and select colors, which are sonified by a synthesis engine. *Esoterion Universe* consists of a 3D space that can be filled with planet-like audiovisual objects, each containing a GUI to modify their visual and sonic properties. *GS.avi* is an instrument that generates continuous spatial visualizations and music from the gestural input of a performer. *Modulant* allows for drawing images, using paintbrush type of tools, which are then sonified.

Six projects were built during Gen.AV 2. *Butterfly* is an audio visualizer which allows for the combination and control of four audio synthesizers, by means of manipulating icons distributed in four XY pads on the screen. *Cantor Dust* generates, displays, and sonifies Cantor set type fractals as sound and visuals. *EUG* further develops *Esoterion Universe* from Gen.AV 1, adding 3D gestural control with a Leap motion sensor. *OnTheTap* plays with the tactile, analog feel of tapping surfaces as interaction input, captured as audio. *residUUm* allows for the creation and manipulation of AV particles, with a variable lifespan, by clicking and dragging on the screen. *Wat* creates a chaotic 3D texture based on cellular automata (Figure 2). All the projects were uploaded to GitHub for download or source-code modification, accessible from `http://www.gen-av.org`. In addition to the code, the project descriptions are available from the same link, facilitating running and replicating the projects.

Expert Interviews

After the performances, the projects were tested for ease of installation and robustness. Six projects were chosen: *Esoterion Universe*, *GS.avi* and *Modulant* from Gen.AV 1; and *Butterfly*, *residUUm* and *Wat* from Gen.AV 2. These were evaluated by expert reviewers, established audiovisual artists who had taken part in the preliminary study interviews. Each expert was given two projects for review, and at least one week time to practice with the software. Thus, each project was evaluated twice (project evaluators E1 and E2). We then conducted semi-structured interviews with the reviewers, lasting an average of 15 minutes per project. They served to follow up on the key issues emerging from the preliminary stage. The questions addressed: 1) the AV content and the relationship between sound and image; 2) ease of use of the software; 3) fluidity, AV manipulability and behavior as an "instrument"; 4) flexibility and reconfigurability of the interface; 5) potential usefulness for other artists and performers.

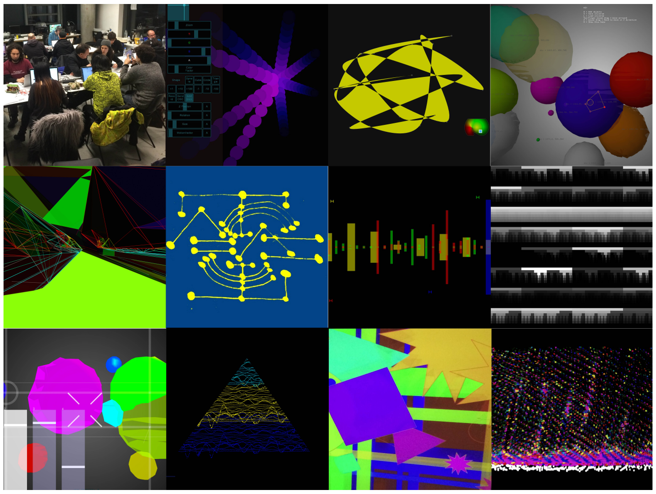

Figure 2. Stage 1: Projects from the hackathons – left to right, then top to bottom: hackathon, ABP, drawSynth, Esoterion Universe, GS.avi, Modulant, Butterfly, Cantor Dust, EUG, OnTheTap, residUUm, Wat.

We conducted a thematic analysis of the interviews. From this analysis, three main themes emerged: *Experience*, *Interfaces* and *Media*. We retained these themes for our analyses of the different stages of the research. Our starting point were Hook's themes related to *interaction*: *constraining interactions* (importance of constraints and focus); *haptically direct* (physical connection); *parallel interaction* (simultaneous control of multiple parameters); *immediacy* (immediate response from the software); *manipulable media* (powerful and varied manipulation of media); *reconfigurable interfaces* (reorganize controls to fit a particular performance); and *visible interaction* (make interaction visible to an audience) [13]. *Immediacy* was a pre-requisite for the selection of the projects. *Constraining interaction* was not detected. We decided to merge *haptically direct* and *parallel interaction*, as they would appear combined in our data. We split *interaction* into two new main themes: *experience* and *interface*. Additional themes related to content emerged, originating the new main theme *media*.

Experience

Visible interaction: Two of the evaluators, in their own practice, prefer not to convey the interface to the audience, and wish to have a separate screen with the GUI for the performer (GS.avi, E1; Modulant, E2), for three main reasons: 1) the visual output could be re-routed without GUI to other software for additional manipulation (GS.avi, E2); 2) the interface is something the audience may not want to see; and 3) having a separate screen would allow for a more complex interface for the performer (Wat, E2). Another evaluator is interested in showing the UI to the audience and "conveying the performer's control on-screen" as part of the experience (residUUm, E2). Taking that approach would allow for "visually reflecting that agency onto the screen", making it understandable for an audience (Butterfly, E2).

Haptic and parallel: The reviewers confirmed the desire for interaction that "provides a sensation akin to being in direct contact or touching and molding media" [13]. The projects did not allow for either haptic or parallel interaction. Multitouch tablets and hardware controllers were mentioned as means to achieve a parallel interaction. Compatibility with hardware controllers and tablets is desired to achieve parallel interaction: controllers would allow for the physicality and "flexibility of an instrument" (Esoterion Universe, E1) and interaction "in a tracking pad on the computer is confusing", with a tablet-based approach being suggested (GS.avi, E2).

Object-oriented: When auditory and visual domains are combined, audiovisual objects can emerge, if simultaneity and a plausible common cause occur [19]. Three projects followed an object-oriented approach, by grouping audio and visual content into distinguishable entities (*Esoterion Universe*, *Butterfly* and *residUUm*). In the first two cases, a GUI was overlaid on the visuals for continuous manipulation. In the third project the opportunity for manipulation occurs only at the genesis of the object. The object approach was considered as being fruitful (Esoterion Universe, E2).

Interface

Reconfigurable interfaces: The reconfiguration of UI becomes possible only by editing the code, which requires specific technical knowledge (Esoterion Universe, E1). Some projects organized the code in order to make it easier to reconfigure: "it's very easy to add your own synths" (Butterfly, E1).

Interface mappings: In some instances, a complex one-to-many mapping of interface to media parameters was considered successful (Butterfly, E2). Evaluators felt that more parameters should be controlled from the UI, resulting in insufficient mapping (Butterfly, E2; Wat, E2). Scalability of layout is desired, as it would allow for additional UI elements (Butterfly, E1).

Interface clarity: The lack of a parameter space in UI elements – an indication of the parameter range, and the current status – was considered problematic in some projects (residUUm, E2).

Interface aesthetics: The gestural aspect of one project was seen as innovative and appealing (GS.avi, E1). It was suggested that it could become more integrated in the visuals by visualizing the gestures (GS.avi, E1). The visual design of the interface is considered to be even more important when conveyed to an audience. In some cases, this design was considered to be unappealing to be shown (Modulant, E2). One reviewer considers that the UI itself should be dynamic, animated in response to the sound (Butterfly, E2). One of the projects adopts a logic of interactive quadrants with XY pads, which was considered to be original and clear (Butterfly, E1).

Media

Manipulable media: Some projects were considered to produce outcomes with a narrow range of diversity (Esoterion Universe, E2; Butterfly, E1). In several cases, the projects rely on 2D or 3D spatial metaphors. There is a desire for an expandable canvas or scene where the media can be presented in and navigated through. This is considered to be missing on one project (Modulant, E2) and praised for its implementation on another (Wat, E1, E2). The satisfactory manipulation possibilities of some projects lead them to be considered "instruments" because of their fluidity (Wat, E1; Modulant, E1).

Generative media: Different evaluators appreciated different degrees of randomness. The generative aspect of some projects was considered to be too chaotic (GS.avi, E1; residUUm, E2). In the balance between generative elements and control, the latter is seen as the priority. But a certain degree of randomness is desirable, and considered to be missing in some cases (Modulant, E1).

Media management: The option to load files in some of the projects is appreciated (GS.avi, E2; Modulant, E1). Runtime loading of content is desired (GS.avi, E1). Real-time sharing of media between applications in the same device, using utilities such as Syphon (http://syphon.v002.info) is wished for (Esoterion Universe, E1). The possibility of accessing networked content is also suggested (Esoterion Universe, E2).

Audience Study

In order to study audience understanding of the performers' actions, we asked audience members of the two performances to fill in a questionnaire about the different projects from Gen.AV 1 and 2 (with the exception of *DrawSynth*, a last minute addition). Respectively 45 and 34 respondents answered the questionnaire. The question asked was: "Did you find the connection between the performer's actions and the audiovisual result understandable?", on a scale of 1 to 5. Projects *Esoterion Universe* and *Modulant* from Gen.AV 1, and *Butterfly*, *EUG* and *residUUm* from Gen.AV 2 obtained the best results (*Modulant* with a median of 5, the others with a median of 4). The five projects that achieved the best results make visible both the interface and the parameter space. *Cantor Dust* and *OnTheTap*, both with a median of 3, implement only one of these aspects (visibility of parameter space in *Cantor Dust*) or only temporarily show them (*OnTheTap*). The remaining projects, with a median of 2, employ neither. These observations informed our design principles.

STAGE 2: AV ZONES CONSOLIDATION PROTOTYPE

Prototype Design

The results from Stage 1 fed into design guidelines for a final prototype, an iPad app for AV performance entitled *AV Zones*. It has been released as open-source (https://github.com/AVUIs/AVZones-beta). It adopts the object-oriented concept of "zones": rectangular areas that incorporate UI elements producing and manipulating sound, and a visualization of that sound. The app has three vertical zones, each with three XY pads for audio manipulation, controlling: pitch shift, delay and filter. Each zone has a sequencer, which can record touch information and visualize it. There are nine sounds available per zone, which can be switched at runtime, and replaced in the code. Different touch inputs create different results: tapping for triggering sounds; touch movement for manipulating the sound; two-finger tap to switch on and off; and double tap to trigger special function – menu or sequencer (Figure 3). The app was developed using the OF environment and the Maximilian audio library (https://github.com/micknoise/Maximilian). Both are open-source and cross-platform.

Initial Tests - Performances

We tested *AV Zones* in "real world" settings: four public performances and two demos in conferences. In a performance, only an iPad is used for audiovisuals. What the performer sees is also what is projected to the audience. The interface is shown on the screen, with touch points being represented by white circles. We made minor improvements between performances. For example, a sequencer was added due to the difficulty of interacting simultaneously with the three zones, and the need to automate some of the processes by recording them.

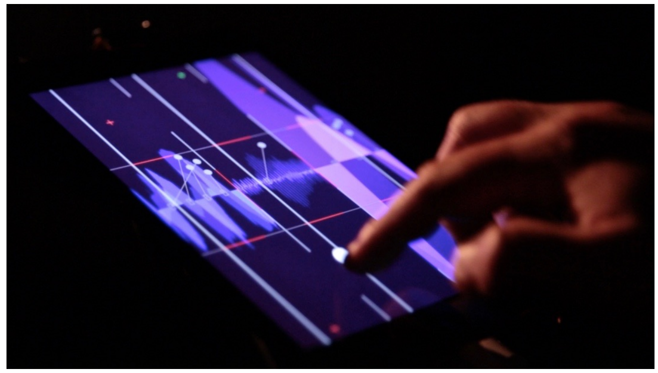

Figure 3. Stage 2: AV Zones prototype.

Expert Interviews

We followed a similar evaluation procedure than in Stage 1: we installed *AV Zones* in iPads and handed them to three evaluators (E1, E2, E3) from our initial expert group. The interviewees tried the app for at least one week. We then ran semi-structured interviews lasting on average 30 minutes. We used the same questions as Stage 1. We ran a thematic analysis, maintaining the three main themes: *media*, *interface* and *experience*. More sub-themes emerged: *sound visualization* (within *media*); *constraining interaction* – a theme that had existed in Hook's analysis but had not appeared in Stage 1; and *playfulness* (within *experience* main theme).

Experience

Constraining interaction: Two evaluators were satisfied with the prototype's design constraints and minimalism, stating that "it's nice to have limitations", having a "minimal simplicity" was pleasing, and its "reduced nature" made it "very appropriate for a live tool" (E1, E3).

Visible interaction: One evaluator would like to be able to hide the UI, completely or partially, and added that by separating what the audience and the performer see, more UI elements could be added on the performer side (E1). Another respondent is satisfied that the audience can see what the performer is doing, as touch points are highlighted with white circles, and would like to see more interactions visualized, such as sound effect manipulation (E2).

Haptic and parallel: One respondent was satisfied with the number of zones and simultaneous control elements (E2), whereas others would like to add an external hardware MIDI controller (E1) or another tablet, creating a dual setup (E3).

Object-oriented: One respondent was pleased with the notion of zones and the way they operate, stating that it was a good concept and design (E2).

Playfulness: The application was considered to be playful – one respondent mentioned multiple times that he enjoyed playing with it, that it was "fun" (E1).

Interface

Interface aesthetics: One evaluator was very pleased with the interface aesthetics of the sequencer functionality, stating that it looks like a "visual music composition" (E1). Another respondent considers that more work could be done in terms of visualizing additional processes in the software, such as loading or choosing sounds (E2).

Interface clarity: The prototype suffers from hidden discoverability issues. Several of the functionalities are activated by different types of touch interaction, not apparent in the UI. Testers had problems activating these, despite the documentation provided (E1, E2, E3). Evaluators complained that it was hard to understand what to do next (E2) leading to getting occasionally stuck (E1).

Reconfigurable interfaces: One respondent in particular was interested in adding reconfigurability options, such as allowing for extending functionalities with software "plugins" that others could build. Another suggestion was having the possibility of grouping zones and nesting them - this would facilitate scaling of zones without overcrowding the screen (E1). One of the respondents suggested adding a back end with substantial configuration options (E3).

Media

Manipulable media: One respondent considers that the software is "a really useful live tool" (E3). The two audio effects, delay and filter, were considered well chosen, and having three simultaneous sounds allows for "enough scope" to maintain a performance (E3). The sequencer is considered an important element for this, as it allows to automate one zone while interacting with others (E1, E2, E3). The prototype is considered to allow for "a different way of approaching sound", less "musical" and "kind of weird" (E1). To have a broader and more musical appeal, two evaluators consider that a stricter timing or "clock" would be important (E1, E3). Having "more authorship and a sense of control" (E1) over the sound is desired. On the visual side, respondents would also like to have greater control. Only one visualization type, with "very little visual configurability" (E3) is considered insufficient. Evaluators would like to be able to have other visualizations and be able to make more choices about them (E1, E2).

Media management: All evaluators would like to be able to load sound files. Although this is possible, it requires modifying the code and re-installing the app, which is inconvenient. The possibility to record sounds is also desired (E2). One evaluator would like to be able to route the visuals to other software for further processing (E1).

Sound visualization: One evaluator considers that the approach followed, to visualize the amplitude levels of the audio buffer, was "fascinating" and "very responsive", particularly at slower speeds (E1). The other two consider this approach to be simplistic, as it does not help to "understand anything about the sound" (E2). They would rather have a "perceptually motivated approach" that would bring it closer to the state of the art (E2) and a two-way interaction between sound and image: not just sound visualization, but also visual sonification (E3).

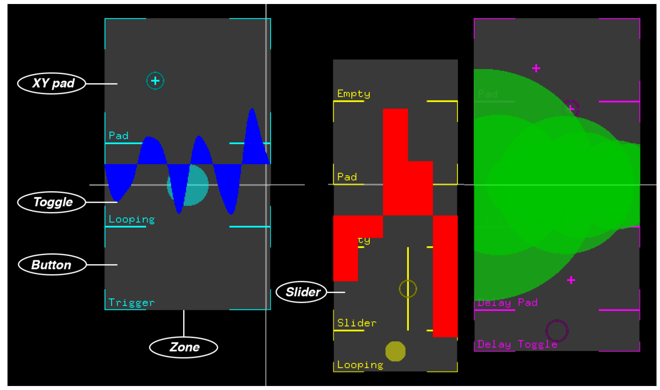

Figure 4. Stage 3: Example of AVUI built with the toolkit, using three zones, with different UI elements and visualizations (explanatory labels in white).

STAGE 3: THE OFXAVUI TOOLKIT

Tookit Design and Development

To assist in making the development of AV work more streamlined, to better integrate interfaces in AV performances, and to make interaction more understandable for audiences, we have developed a toolkit for combining UI with AV content. We generalized knowledge gained from the previous stages in the development of ofxAVUI, a modular, reusable software toolkit to facilitate the production of audiovisual user interfaces. The evaluation of the 11 Stage 1 prototypes and the app from Stage 2 were distilled into a set of design insights, which in turn contributed to the definition of the feature set of our toolkit. This led to a technical specification and software architecture. The design specifications for the toolkit were, divided by main themes: 1) experience – allow for parallel and visible interaction; integrate sound, image and UI following an object-oriented approach; 2) interface – enable reconfigurable interfaces, with flexible mappings; ensure both clarity and aesthetic appeal of interface, harmonized with visuals; 3) media – allow for powerful media manipulation, with procedural content; and adopt a flexible media management. Full design specifications, and their connection with previous stages of the research can be found at `http://www.gen-av.org/avui-design-tables/`.

For the development of the toolkit, we again used OF and Maximilian. We organized the code into three groups of class files: audio, visuals and UI. Each of the three groups has a base class, making it easy to extend and to create a new audio process, a new visualization and a new UI type. The style of the UI is centralized in one class, facilitating the customization of its appearance. It was released as an "add-on" (plug-in for OF), allowing to be integrated in other OF projects by developers. We released the add-on in versions for personal computer and mobile multitouch devices. ofxAVUI was released as open source in our GitHub repository (`https://github.com/AVUIs/ofxAVUI`). As is customary with OF add-ons, we included examples, extensively commented the source code, and adopted the "ofx" prefix. It is now part of the main directory for OF add-ons, in the UI category: http://ofxaddons.com/categories/1-gui.

We kept the object-oriented notion of zones from the app in Stage 2, as an organization structure for combining AV and UI. Each 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 defined, as well as their: size; position; color palette; UI elements. Any parameter from the UI can be rerouted to any audio feature of the zone, or any other aspect of the software (for example, any graphic on the screen). We kept the minimal UI aesthetics of the prototype. Visualization is an important link between sound and image, therefore we added two visualizations, with more configuration options. We also facilitated the creation of new visualizations, making the visualization module extensible. We incorporated the Syphon protocol, so that media could be channeled, with or without UI, to other applications (Figure 4). These design elements, core to the definition of an AVUI, are exposed to the OF developer through high level function calls, making integration into an OF project straightforward.

As an example of ease of ofxAVUI implementation, only three lines of code are needed to create and configure an AV zone with a button that triggers a sound and associated visualization. UI and visualization inherit the aesthetic properties configured for the zone.

Evaluation

For a first, internal validation of ofxAVUI, we built a general release version of our Stage 2 prototype using the add-on. This new version allowed us to address areas to improve in *AV Zones* identified during Stage 2: interface clarity, media manipulation, and media management. In terms of interface clarity, we separated the multiple functions of the XY pad into dedicated toggles and buttons (on/off toggle, sequencer toggle, sound file drop-down menu). Regarding media manipulation, the sequencer can now record and visualize additional interactions. As for media management, users can add and manage sound files via the Apple iTunes interface, a standard for iOS apps. This final version of *AV Zones* has been released as open source on GitHub (https://github.com/AVUIs/AVZones-ofxAVUI), and can be loaded into an iOS device using Apple's Xcode software. The ofxAVUI add-on allowed us to easily and quickly redevelop our prototype and solve issues detected in Stage 2 (Figure 5).

To evaluate the add-on with other developers, we organized a one-day hackathon to look at its ease of use and effectiveness of development. A call was circulated using the same channels as the Stage 1 hackathons. Eight participants took part in the hackathon (five male, three female). Their profile was similar to the previous participants: audiovisual performers and developers. Four of the participants managed to complete a small project during the one-day event. The projects were: *FFT/MFCC*, audio frequency analyzers and visualizers; *Step Sequencer* for creating rhythmic patterns; *Background Image*, for customizing zones; and *Lisajous and Grid*, two additional

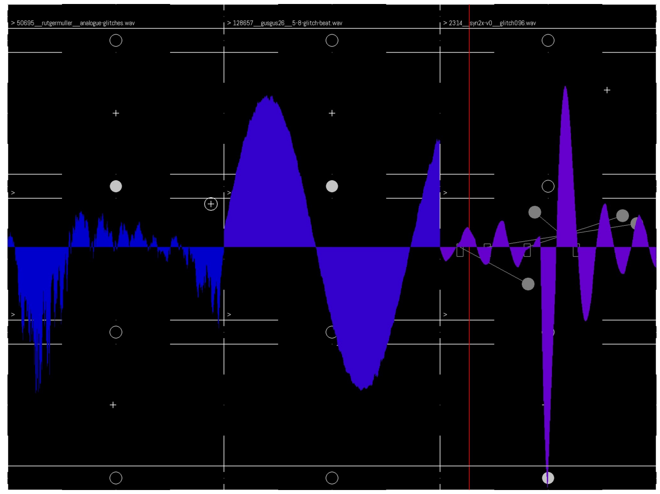

Figure 5. Stage 3: AV Zones, rebuilt with ofxAVUI.

visualizers. These projects expand the toolkit and were added to the ofxAVUI online repository (Figure 6).

In order to obtain further feedback, we reached out to ofxAVUI users on GitHub. Although software downloads are anonymous, 12 individuals had "starred" the repository – a form of following the repository and its updates on GitHub. Of those 12, eight had contact information in their GitHub profiles and were contacted by us. We sent an email asking if they would like to participate in a study. We obtained four replies, and two developers agreed to participate. They developed two projects: a four-zone *Multisampler* and *ShaderUI*, an implementation of sound-responsive shaders. They were also added to our repository.

We conducted face-to-face interviews with the participants in the hackathon (E1-8), and Skype interviews with the two online developers (E9, E10). Interviews lasted on average 30 minutes. The semi-structured interviews addressed: 1) ease of development with ofxAVUI; 2) its usefulness; 3) the appeal of its design; 4) results achieved and satisfaction with those; 5) potential for future use of the add-on.

In our thematic analysis of the interviews, a new main theme emerged: *development*, related to observations on programming and code. From this, three sub-themes were considered: *organization and architecture of code*, *speed and ease of development*, and *patching and building*. This main theme preempted the *reconfigurable interfaces* theme of the previous studies. One additional theme emerged: *scenarios*, under the *experience* main theme.

Development

Organization and architecture of code: Most respondents considered that the code was well organized, with "everything nicely in their respective categories" (E1), and that it was easy to see "how the objects related to each other" (E9). Some evaluators mentioned that the code was easy to extend, as every category has a base class (E3, E10). One respondent highlighted the flexibility in mapping UI parameters to other

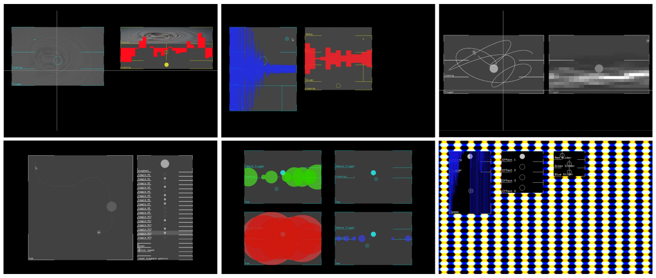

Figure 6. Stage 3: Projects from the ofxAVUI hackathon (first four) and online evaluators (last two) – left to right, then top to bottom: *Background Image*, *FFT/MFCC*, *Lisajous and Grid*, *Step Sequencer*, *Multisampler* and *ShaderUI*.

zones. Different from other UI toolkits, ofxAVUI is, according to our users, designed "to reuse bits and pieces in particular ways to invent new stuff" (E10). Three respondents wanted more abstraction in the code (E2, E8, E9): flexibility regarding multiple types of input (for example, touchscreen or sensors), and to be able to "switch audio engines at will" (E9). One evaluator could not finish the project on time because the toolkit did not support PureData (https://puredata.info) as audio engine (E8). Two evaluators felt that the UI could have taken advantage from the existing ofxGUI toolkit (E1, E3).

Speed and ease of development: ofxAVUI was considered easy to work with by respondents (E3, E4, E7, E9, E10), not just because of its organization and architecture, but also because "it already has the minimum package of sound, UI and visual" built in (E3). One of the respondents considers ofxAVUI easier to use than the two main UI toolkit for OF, ofxUI and ofxGUI (E10). This evaluator considers it easier and faster to prototype with than using related tools, and that "it fills a gap", providing "interesting opportunities that would not be as easily possible previously". Two respondents thought that the add-on could be better documented (E1, E10).

Patching and building: Three respondents would like to have a top-level environment that facilitates the creation of UIs, with a simplified coding language (E1, E3, E7), or by drag-and-drop, as in visual interface builders (E1). Two evaluators expressed interest in having a "master controller" (E5, E10) that could switch zones on and off, and reroute information between them. Another was interested in integrating zones and patching them (E3). Yet another suggested having multiple visualizations per zone, stacked in layers, visualizing different parameters (E5). Two respondents are interested in the implementation of the OSC protocol (http://opensoundcontrol.org) to control other applications and devices (E6, E10), with one stating that it was faster to build a OSC controller with ofxAVUI than with Lemur, a popular tablet controller builder app (E10).

Experience

Visible interaction: In a use case that involves showing ofxAVUI to an audience, such as a performance or a demo, most respondents (E1, E3, E5, E6, E8, E10) consider that revealing the UI is important, making the software "engaging" and "easy to understand". It creates "a more cohesive experience", by showing "the beauty of the internals of the system" (E3). One respondent considers that showing an UI would depend on the use case, and that revealing it "challenges how you interact" in order to "find ways to make audience aware of what's happening" (E5).

Scenarios: Two evaluators considered ofxAVUI well suited to teach sound and visualization (E2, E8). One respondent considered ofxAVUI adequate for game development, particularly pedagogical games (E2). He also suggested that ofxAVUI could be used in more generic applications, for highlighting important tasks. Another evaluator stated that the adoption of OF makes the toolkit more suited to artistic applications, but if made more "portable, or not relying on OF" it could be used for more "day-to-day software development" (E9).

Interface

Interface clarity: Two evaluators requested more visual feedback for changes of state in the UI (E2, E6), such as hovering.

Interface aesthetics: Several respondents liked the minimalist "bare bones" aesthetics of the UI elements (E1, E3, E9). One evaluator wanted to customize UI elements and implement UI "themes", and developed a project for adding background images to zones (E1). One respondent mentioned that the large size of the UI elements "seems more applicable to a touch interface than a mouse interface" (E10).

Media

Sound visualization: One respondent wanted to have more possibilities for audio analysis, and created a project in that direction, based on frequency visualizers. He suggested that more audio information retrieval techniques and 3D visualizations could be added (E2). One respondent wanted to have not only sound visualization, but also visual sonification (E1). Another considered that the visualizations should be used not just for sound, but to visualize other data (E10).

Media manipulation: One evaluator (E4) showed interest in having live audio input. Another respondent (E5) wanted to synthesize sound from an image and its color information. One respondent was interested in rhythmical and quantized aspects of sound, and developed a step sequencer (E3).

AVUI GUIDELINES

The best practices identified in our research allow us to propose the following design guidelines for AVUIs, for use by designers who wish to implement AVUIs, either using ofxAVUI or a different approach. They may be useful for designers who wish to use sound and image together in the interface, either by using ofxAVUI or by using different technologies. These guidelines are divided into three topics, which match the three main themes across the different stages:

1) Maximizing AV Experience

a) Develop AVUIs that can be implemented across multiple platforms and interaction modalities – multitouch interaction seems particularly suited, as it allows for the synaesthetic illusion of touching and molding the audiovisuals;
b) Consider the potential of AVUIs for facilitating visualization of interaction when sharing/showing a screen;
c) Adopt an object-oriented approach, for a harmonious,

coherent and interrelated convergence of audio, image and UI;
d) Facilitate different types of display, allowing for different performer-audience display configurations and hardware.

2) Optimizing Interface Functionality and Aesthetics

a) Use reconfigurable interfaces, possibly with a back-end, that allow to re-map elements of the UI to different sonic features and visual properties; that can also change how the sound is visualized; and that can have an extensible architecture in order to better allow for customization;
b) Explore not simply one-to-one but also one-to-many mappings between UI, audio and visual features;
c) Adopt a minimalist interface aesthetics that integrates well with the visuals, namely regarding color, shape and movement, and that does not detract from the visuals;
d) Reinforce interface clarity by ensuring visibility of all UI elements, their state, parameter space and current position to it; and verify that the visualizations do not detract from this;
e) Allow for hierarchical interfaces, with the possibility of a master control, and communication between modules.

3) Media Strategies

a) Allow for powerful manipulation of sound and image: different forms of media generation, such as different forms of sound and visual synthesis and sampling; multiple audio and visual effects; and experiment with mappings between UI, audio and visuals across different properties;
b) Make use of generative media, due to its variety, flexibility and economy of resources;
c) Try different visualization and sonification approaches, using information retrieval techniques from audio and image;
d) Visualization should reflect not only audio but also the multiple interactions afforded by the UI;
e) Leverage powerful media management features, such as networked content (for example, streamed audio or visuals), audio and visual input, and content sharing between applications in the same device.

DISCUSSION

Our multi-stage research produced a range of concrete outcomes: prototypes by us as well as by participants; a software plug-in toolkit; and an app built with that plug-in. This process allowed us, in an iterative user-centric manner, to gain insight on AVUI design, summarized as a set of AVUI design guidelines above, with implications for design discussed here.

Comparison with Existing Solutions

Since ofxAVUI is built in C++ and with the popular openFrameworks toolkit, it can be easily adopted by digital artists and designers, and embedded in other C++ code (without having to resort to OSC, although it also supports it). Stage 3 evaluators were very positive regarding speed and ease of development with ofxAVUI, compared to other solutions (such as ofxUI and ofxGUI), and most of them consider it easy to use. One of the main trade-offs of ofxAVUI versus other solutions is, due to its inherent pre-packaging of AV and UI, it is not as flexible as using assorted graphics and sound toolkits to build a solution from scratch. On the other hand, it is quicker, and already establishes a harmonization of UI with content.

Object-oriented Integration of Interface with AV

In an AVUI as we propose, sound, visualization and user interface are integrated, functionally and aesthetically, into the same entity. This relates to the concept of audiovisual objects in cognitive science [19]. The results from our studies confirm the appeal of an object-oriented approach to AV interaction design [6]. By analyzing sound and representing it visually in real-time, sound and image are harmonized, synchronous and coherent. Audiovisual congruency is ensured, making use of the identified benefits of crossmodal congruency regarding task accuracy and engagement [5, 11, 25, 31]. This object-oriented approach to AVUI design is apt to situations where the display and interaction plane are fused, as in multitouch displays, allowing for "a sensation akin to being in direct contact or touching and molding media" [13].

Visibility of Interaction

The visibility of interaction is inherent in an AVUI: an interaction triggers either a visual or a sound, which is visualized. An AVUI can be particularly suited for use cases where there are benefits from representing user interaction, such as: performances (our case-study); remote collaboration and telepresence; presentations and demos. However, we have detected different profiles of users regarding visible interaction: some prefer to visually reflect agency onto the screen, "making it understandable for an audience" (Butterfly, E2), others consider that audiences do not necessarily want to see agency on the screen. In functional applications, this will be determined by the task at hand. For creative applications, this can be a matter of taste, as we noted with our expert evaluators in Stage 1. In either case, the AVUI allows the developer to merge visual content with interface elements.

Reconfigurable Interfaces and Flexible Mappings

One of the identified strong points of ofxAVUI was its modularity and reconfigurability: it is designed to recombine UI and media in different configurations and mappings, providing "interesting opportunities that would not be as easily possible previously" (user E10). In ofxAVUI, being able to easily route any UI parameter to any aspect within a zone (sound, visuals, other UI elements) or outside the zone (for example, any graphic on the screen) was considered by our users as innovative with respect to other UI toolkits. Flexibility of mappings between UI, sound and image, and ways to manage that flexibility, are fundamental qualities of an AVUI, and important features in enabling technologies for them to be useful to interaction designers across a range of application domains.

Design Constraints and Speed of Development

User E3 appreciated that ofxAVUI provides the core necessary functionality in sound, UI, and visuals. This is considered an advantage for some: one evaluator considers that he can "prototype a certain part of my process faster" with ofxAVUI, and that it is easier to use than other UI toolkits (E10). For others, ofxAVUI is too constrained precisely because it is tied to specific packages of sound, UI and visuals. These evaluators would like more abstraction, to be able to replace certain elements of the toolkit (for example, the audio library).

Some evaluators would like ofxAVUI to be simpler to use, for example by adding a GUI layer that would allow users to build AVUIs by dragging and dropping elements, as in traditional interface builders. Therefore, there seems to be a desire for both higher level ease of use, and lower level flexibility. A better balance could be pursued in the future between ease of use by pre-packaging elements, and allowing for more architectural flexibility. Having been built with OF is a constraint in itself – OF is popular in media art and design, but not used as much for more generic development. The toolkit could be ported to other frameworks to facilitate its adoption for more generic use cases.

Participatory Design and Hackathons

Our participatory design approach enabled us to leverage artistic knowledge in audiovisual performance from a range of practitioners into a generic software toolkit. AV performers are specialists in sound visualization and visual sonification – audiovisual crossmodal interactions. We believe that their AV design skills were an important contribution to this research that benefited the design of the ofxAVUI technology. Additionally, these users make high demands for an AV system in terms of media manipulation and interaction design: they can be considered super-users, who are regularly performing in front of an audience, and need powerful, fluid and responsive manipulation of AV media through a robust interface.

We used hackathons as a motivating, productive way to connect with our users. Hackathons were employed from two different perspectives. In Stage 1, two hackathons were used for the rapid prototyping of AV performance systems by AV artists. In Stage 3, the hackathon aimed to test in a short period of time the ease of implementation of our toolkit. Participants were asked to develop a project with it, which could be added to our toolkit repository, as an extension of its functionality or a demonstration of a new use case. We also reached out to the community of interaction designers and developers following our GitHub repository website. In this sense, we complemented the "local" perspective of the hackathons with the "global" community of GitHub users, adapting, albeit on a smaller scale, the approach followed in [28].

Multi-stage Approach

Informed by related multi-stage studies [33, 35], we adopted a three-stage approach to the development of the ofxAVUI toolkit. This could be used for the development of other technologies. It can be summarized as *explore-consolidate-abstract* approach: 1) explore and gather multiple views via participatory design process, and evaluate results with other users; 2) design a prototype that consolidates best practices detected in the previous study, and evaluate with users; 3) develop a general technology based on the evaluation of the prototype, convert and abstract positive aspects of it into a toolkit, and run an additional participatory design and evaluation session for testing. This approach, with a participatory stage based on hackathons, allowed us to iteratively develop both our AVUI Guidelines and the ofxAVUI toolkit. Conclusions from each study were converted into design specifications for the following one. Conclusions from each stage fed into design specifications for the following one, with the last stage informing a set of general design guidelines.

AVUI as Parallel to Crossmodality in the Real World

Our interactions with the world are multi-sensorial. Opening a door handles produces auditory and visual feedback. Some of these interactions, such as pouring water into a glass, give us audiovisual feedback regarding dimensional data. In these interactions, audio and visual information are related in a congruent way. The concept of auditory icons aims to "to use sound in a way that is analogous to the use of visual icons to provide information," providing a a natural way to represent dimensional data as well as conceptual objects [10]. AVUI extends Gaver's pragmatic concept by proposing a crossmodal approach that incorporates UI elements, sonic feedback and congruent visualization in a way that aesthetic content and interface become one. The integrated audio and visualization reflect the status of UI elements, recalling a functional simplicity of the sort encouraged by John Maeda [22].

CONCLUSION

We have introduced the concept of AudioVisual User Interface (AVUI), a type of interface where UI, audio and visualization are interconnected and integrated. By combining UI with interrelated sound and image, the proposed concept of AVUI (and ofxAVUI toolkit in particular) can help leverage the identified benefits of audiovisual crossmodal interaction, such as improvements in performance and engagement.

We presented an iterative multi-stage process of design, prototyping, development and evaluation of ofxAVUI, an enabling software toolkit to facilitate development of AVUIs. The toolkit has been released as open-source, and is multi-platform, aiming to facilitate its adoption. Participatory design methods were used, centered around three hackathons. This process also allowed us to incorporate expert and practitioner insight into a series of generic guidelines for the design of AVUIs. The toolkit and guidelines will be of interest to interaction designers who wish to create compelling products and systems which integrate sound and image in the interface. By extending Hook's existing theoretical framework to study a large number of AV systems, we believe that the findings have a strong generalizability that the previous studies do not.

We believe that the AVUI concept and the ofxAVUI technology have potential for application in a number of use cases where a screen and interaction is shown to an audience, and for and multimodal interaction. The crossmodal linkages that an AVUI facilitate could be useful for engagement in VR and AR interactive environments. This form of interaction which fuses sensing modalities, function and content, can be compelling for a number of domains: not only areas where engagement is important, such as art, education and games, but also assistive and accessible technologies.

ACKNOWLEDGEMENTS

We would like to thank all who collaborated in this research: workshop and hackathon participants; evaluators and interviewees; colleagues; and in particular Borut Kumperščak for programming assistance and Alessandro Altavilla for documentation. This work was supported by the EU Marie Curie fellowship FP7 REA grant 627922 and also partly by FCT/MCTES LARSyS (UID/EEA/50009/2013 (2015-2017)).

REFERENCES

1. Paul Bach-y Rita. 1987. Brain Plasticity as a Basis of Sensory Substitution. *Neurorehabilitation and Neural Repair* 1, 2 (June 1987), 67–71. DOI: http://dx.doi.org/10.1177/136140968700100202
2. Florent Berthaut, Mark T. Marshall, Sriram Subramanian, and Martin Hachet. 2013. Rouages: Revealing the Mechanisms of Digital Musical Instruments to the Audience. In *Proceedings of New Interfaces for Musical Expression (NIME)*. Daejeon, Seoul. http://nime.org/proceedings/2013/nime2013_51.pdf
3. Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative Research in Psychology* 3, 2 (2006), 77–101. DOI: http://dx.doi.org/10.1191/1478088706qp063oa
4. Ana Carvalho and Cornelia Lund (Eds.). 2015. *The Audiovisual Breakthrough*. Fluctuating Images, Berlin. http://www.ephemeral-expanded.net/audiovisualbreakthrough/
5. Karen Collins and Peter J. Taillon. 2012. Visualized sound effect icons for improved multimedia accessibility: A pilot study. *Entertainment Computing* 3, 1 (Jan. 2012), 11–17. DOI: http://dx.doi.org/10.1016/j.entcom.2011.09.002
6. Nuno N. Correia. 2013. *Interactive Audiovisual Objects*. Doctoral thesis. Aalto ARTS Books, Helsinki. https://www.taik.fi/kirjakauppa/product_info.php?cPath=26&products_id=269&language=en
7. Nuno N. Correia and Atau Tanaka. 2014. User-Centered Design of a Tool for Interactive Computer-Generated Audiovisuals. In *Proc. 2nd ICLI*. Lisbon. http://www.liveinterfaces.org/proceedings2014.html
8. Bram Crevits. 2006. The Roots of VJing. In *VJ: Audio-Visual Art and VJ Culture*, Michael Faulkner (Ed.). Laurence King Publishing, London, 14–19.
9. Carl DiSalvo and Melissa Gregg. 2013. Issue-Oriented Hackathons. (2013). http://carldisalvo.com/posts/issue-oriented-hackathons/
10. William W. Gaver. 1986. Auditory Icons: Using Sound in Computer Interfaces. *Hum.-Comput. Interact.* 2, 2 (June 1986), 167–177. DOI: http://dx.doi.org/10.1207/s15327051hci0202_3
11. F. Wai-ling Ho-Ching, Jennifer Mankoff, and James A. Landay. 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)*. ACM, New York, NY, USA, 161–168. DOI: http://dx.doi.org/10.1145/642611.642641
12. Lars Erik Holmquist. 2008. Bootlegging: Multidisciplinary Brainstorming with Cut-ups. In *Proceedings of the Tenth Anniversary Conference on Participatory Design 2008 (PDC '08)*. Indiana University, Indianapolis, IN, USA, 158–161. http://dl.acm.org/citation.cfm?id=1795234.1795260
13. Jonathan Hook, David Green, John McCarthy, Stuart Taylor, Peter Wright, and Patrick Olivier. 2011. A VJ centered exploration of expressive interaction. In *Proc. CHI'11 (CHI '11)*. ACM, New York, NY, USA, 1265–1274. DOI: http://dx.doi.org/10.1145/1978942.1979130
14. Jonathan Hook, John McCarthy, Peter Wright, and Patrick Olivier. 2013. Waves: exploring idiographic design for live performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 2969–2978. DOI: http://dx.doi.org/10.1145/2470654.2481412
15. Michael Xuelin Huang, Will W. W. Tang, Kenneth W. K. Lo, C. K. Lau, Grace Ngai, and Stephen Chan. 2012. MelodicBrush: A Novel System for Cross-modal Digital Art Creation Linking Calligraphy and Music. In *Proceedings of the Designing Interactive Systems Conference (DIS '12)*. ACM, New York, NY, USA, 418–427. DOI: http://dx.doi.org/10.1145/2317956.2318018
16. Lilly Irani. 2015. Hackathons and the Making of Entrepreneurial Citizenship. *Science, Technology & Human Values* (April 2015), 1–26. DOI: http://dx.doi.org/10.1177/0162243915578486
17. Sergi Jordà. 2003. Sonigraphical Instruments From FMOL to the reacTable*. In *NIME03*. Montreal. http://www.nime.org/proceedings/2003/nime2003_070.pdf
18. M. Komssi, D. Pichlis, M. Raatikainen, K. Kindstrom, and J. Jarvinen. 2015. What are Hackathons for? *IEEE Software* 32, 5 (Sept. 2015), 60–67. DOI: http://dx.doi.org/10.1109/MS.2014.78
19. Michael Kubovy and Michael Schutz. 2010. Audio-Visual Objects. *Review of Philosophy and Psychology* 1, 1 (March 2010), 41–61. DOI: http://dx.doi.org/10.1007/s13164-009-0004-5
20. Golan Levin. 2000. *Painterly Interfaces for Audiovisual Performance*. Ph.D. Dissertation. M.Sc. Dissertation. Massachusetts Institute of Technology, Cambridge, MA. http://acg.media.mit.edu/people/golan/thesis/thesis300.pdf
21. Thomas James Lodato and Carl DiSalvo. 2016. Issue-oriented hackathons as material participation. *New Media & Society* (Feb. 2016), 1461444816629467. DOI: http://dx.doi.org/10.1177/1461444816629467
22. John Maeda. 2006. *The Laws of Simplicity*. The MIT Press, Cambridge, MA.
23. Thor Magnusson. 2006. Affordances and Constraints in Screen-based Musical Instruments. In *Proceedings of the 4th Nordic Conference on Human-computer Interaction: Changing Roles (NordiCHI '06)*. ACM, New York, NY, USA, 441–444. DOI: http://dx.doi.org/10.1145/1182475.1182532

24. Harry Mcgurk and John Macdonald. 1976. Hearing lips and seeing voices. *Nature* 264, 5588 (Dec. 1976), 746–748. DOI:http://dx.doi.org/10.1038/264746a0
25. Oussama Metatla, Nuno N. Correia, Fiore Martin, Nick Bryan-Kinns, and Tony Stockman. 2016. Tap the ShapeTones: Exploring the Effects of Crossmodal Congruence in an Audio-Visual Interface. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 1055–1066. DOI: http://dx.doi.org/10.1145/2858036.2858456
26. William Moritz. 1997a. Digital Harmony: The Life of John Whitney, Computer Animation Pioneer. 2.5 (Aug. 1997). http://www.awn.com/mag/issue2.5/2.5pages/2.5moritzwhitney.html
27. William Moritz. 1997b. The Dream of Color Music, And Machines That Made it Possible. *Animation World Magazine* 2.1 (April 1997). http://www.awn.com/mag/issue2.1/articles/moritz2.1.html
28. Alistair Morrison, Donald McMillan, Stuart Reeves, Scott Sherwood, and Matthew Chalmers. 2012. A Hybrid Mass Participation Approach to Mobile Software Trials. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 1311–1320. DOI: http://dx.doi.org/10.1145/2207676.2208588
29. Paul Nagle. 2008. Yamaha Tenori-On Review. *Sound On Sound* Feb. 2008 (2008). http://www.soundonsound.com/sos/feb08/articles/yamahatenorion.htm
30. I. Olowe, G. Moro, and M. Barthet. 2016. residUUm: user mapping and performance strategies for multilayered live audiovisual generation. In *Proceedings of International Conference on New Interfaces for Musical Expression (NIME)*. Brisbane. http://www.nime.org/proceedings/2016/nime2016_paper0053.pdf
31. Shin'ichi Onimaru, Taro Uraoka, Naoyuki Matsuzaki, and Michiteru Kitazaki. 2008. Cross-modal information display to improve driving performance. In *Proc. ACM symposium on Virtual reality software and technology (VRST '08)*. ACM, New York, NY, USA, 281–282. DOI: http://dx.doi.org/10.1145/1450579.1450653
32. Chris Salter. 2010. *Entangled: Technology and the Transformation of Performance*. MIT Press, Massachusetts.
33. Guy Schofield, David Green, Thomas Smith, Peter Wright, and Patrick Olivier. 2014. Cinejack: Using Live Music to Control Narrative Visuals. In *Proceedings of the 2014 Conference on Designing Interactive Systems (DIS '14)*. ACM, New York, NY, USA, 209–218. DOI: http://dx.doi.org/10.1145/2598510.2598520
34. Ladan Shams, Yukiyasu Kamitani, and Shinsuke Shimojo. 2000. Illusions: What you see is what you hear. *Nature* 408, 6814 (Dec. 2000), 788–788. DOI: http://dx.doi.org/10.1038/35048669
35. Alexander Wiethoff and Sven Gehring. 2012. Designing Interaction with Media Façades: A Case Study. In *Proceedings of the Designing Interactive Systems Conference (DIS '12)*. ACM, New York, NY, USA, 308–317. DOI: http://dx.doi.org/10.1145/2317956.2318004