The Smart Stage: Designing 3D interaction metaphors for immersive and ubiquitous music systems

Francisco Bernardo, Pedro D. Pestana and Luís G. Martins

Abstract—**This conceptual paper describes a work in progress in the process of design and implementation of the Smart Stage, an interactive music system prototype for collaborative musical creativity in immersive and ubiquitous environments. This system is intended to have a low entry barrier, thus more forgiving to users with lesser experience or knowledge in music, and it is designed with affordances to support intuitive progress in improvisational performance in a collaborative setting. We present a preliminary technical overview of the system and a first case study of a 3D interaction metaphor for granular synthesis, developed for this environment.**

Index Terms—**3D User Interfaces, Interactive Music Systems, Multimodal Gestural Acquisition, Prototyping, Ubiquitous Computing.**

I. Introduction

This paper focuses on the description of the design process of an interactive music system for creative and collaborative music making. We envision the Smart Stage, firstly, as a ubiquitous system for musical performance in immersive spaces that brings pervasive musical expression to a wider non-expert audience, building upon creativity, collaboration and social interaction; secondly, as a tool that supports the investigation of digital music interaction through research on the design of meaningful interaction metaphors and innovative interfaces for real-time music performance and improvisation.

We aim for designs that support intuitive progress in improvisational performance based in multimodal gestural acquisition in a collaborative setting. Therefore, these interaction metaphors and interfaces should embed affordances (defined by Gibson (1978) as objective, actionable properties of objects that are perceivable as such) that provide both a low entry barrier and a high ceiling in interactive music systems. They should be more forgiving to non-expert users, but engaging in the long term to enable performers to develop their skills.

Inspired on the work of Johnson and Larson (2003) on musical motion metaphors, we build upon their concepts as a framework for interface design and interaction for control of music and audio processes. We present a first case study of a 3D multimodal interaction metaphor for the control of granular synthesis (GS) through motion capture in spatial augmented reality (SAR) settings.

This paper is structured as follows. Section 2 discusses the related work in the research areas that informs our design process. Section 3 documents requirements and specifications for system design. Section 4 makes an overview of our implementation proposal of the Smart Stage system and a specific 3D music interaction metaphor. In section 5, we reflect and discuss our proposal. Finally, we indicate future perspectives and conclude the paper in section 6.

II. Related Work

The ideation and design process has been informed by recent efforts in areas of research such as ubiquitous computing, interaction design, augmented reality and musical metaphors.

A. From Ubiquitous Computing to Ubiquitous Music Systems

As UbiComp departed from the original vision of Marc Weiser (1991) and blossomed throughout the last decades, it spawned new research areas such as Ambient Intelligence (AmI), Context-Aware Applications, and Mobile and Pervasive Computing [4]. These areas focused on the development of systems with user-centric design, context awareness, user monitoring and tracking supported by sensor integration, and an application layer with post-WIMP 1 interfaces. Recent research in UbiComp claims, however, the need for new directions in this field. Rogers (2006) proposes a new, broader and more attainable agenda focused on the design of new engaging user experiences that catalyse creativity through playful and learning practices.

UbiComp has converged with music in a young research field designated by Ubiquitous Music (Pimenta et al, 2009). It encompasses UbiComp technology and concepts [3], the fields of mobile music (Tanaka, 2004; Essl & Rohs, 2009;

Manuscript received January 16, 2015. F. Bernardo thanks the Innovation Agency (Agência de Inovação, ADI, Portugal) and *Quadro de Referência Estratégico Nacional* (QREN, Portugal), for funding this work through project *VisualYzARt: Visual programming framework for augmented reality and ubiquitous natural user interfaces* (QREN-ADI ref: 23201) and *COMPETE - Programa Operacional Factores de Competitividade* (POFC).

F. Bernardo is with the Research Center for Science and Technology in the Arts (CITAR), Porto, Portugal (e-mail: fbernardo@porto.ucp.pt).

P. Pestana is with the Research Center for Science and Technology in the Arts (CITAR), Porto, Portugal (e-mail: ppestana@porto.ucp.pt)

L. G. Martins is with the Research Center for Science and Technology in the Arts (CITAR), Porto, Portugal (e-mail: lmartins@porto.ucp.pt)

[1] Van Dam (1997) introduced the concept of post-WIMP user interfaces, referring to interfaces that do not use menus, forms, or toolbars, but that rely on "gesture and speech recognition for operand and operation specification".

Flores et al., 2010) networked music [10], and involves open, participative and non-trivial musical practices.

Ubiquitous music systems implement this concept, as musical computing environments that support multiple users, devices, sound sources and activities in an integrated way, providing mobility, social interaction, device independence and context awareness (Pimenta et al, 2009). The emergence of collaborative systems for music making that build upon some of these fundamental premises is becoming an evident trend in recent research (Blaine & Fels, 2003; Barbosa, 2003; Kaltenbrunner, Jorda, Geiger, & Alonso, 2006; Miletto et al., 2011; Morreale et al., 2013). These systems seem to share some common traits such as strong restringements concerning musical control, and precedence of the engagement and overall experience over the generation of music. They are also associated to a common user profile that seeks new purposes, like engagement, entertainment and self-expression, and that may be turned into an active music producer if provided with a supporting system.

Under the HCI perspective, one must strive to identify and characterize these roles, their goals, and the tasks they need to perform to achieve them, in order to inform the successful design of such systems. Expert music interfaces are designed to offer a wide range of expression and long path to virtuosity, to which Blaine and Fels (2003) refer to as the "pathway to expert performance". These interfaces should support real musical performance, which Wanderley and Orio (2002) define as the continuous changes of sound parameters exerted by a controller. In that sense, a controller must afford accuracy, resolution, range of perceived features, and support musical tasks that should strive to attain temporal precision so that musicians have complete temporal control of the performance parameters [14]. Intuitively, it is expected that interfaces for the novice role be designed with different affordances than those of expert music interfaces. In fact, most of expert interfaces features are only suitable for musicians, not for novices. Nevertheless, designing interactive systems that include support for both expert and non-expert user roles poses a great challenge, given the difficulty of the trade-off that has to be achieved.

B. Immersive Spaces and 3D Music Interaction

The use of augmented reality and other mixed reality media for artistic purposes goes back to 1969 with Videoplace, a seminal work of AR art developed by Myron Krueger (1977). Krueger created methods to provide rich sensory experiences through simple means based on cognitive strategies of augmentation and transformation, placing emphasis on silhouettes and synaesthesia (Vajpeyi, 2001). Videoplace offered the possibility to program series of simulations and actions, and provided over 50 compositions and interactions, including Critter, Individual Medley, Fractal, Finger Painting, Digital Drawing, Body Surfacing, Replay, among others. More recently, Golan Levin (2000) among others have presented research and artistic works around immersive environments for performance with a strong relation with music.

The adoption of AR and 3D interaction technologies for specific uses in music has been motivated by different goals, spanning fan engagement, marketing and promotion, live show enhancement, and of course, the creation of new interfaces for music making. Mulder and Fels (1998) introduce Sound Sculpting, an environment for designing and performing virtual musical instruments with 3D geometry. This environment allows interaction through the use of Cybergloves and Polhemus sensors and implements manipulation pragmatics such as carving, chiselling, claying and assembling of Virtual Objects (VO), that map onto sound space in the of sound effects such as flange strength, chorus depth, FM distortion and vibrato.

Poupyrev, Berry and Kurumisawa (2000) introduce Augmented Groove as a musical interface that explores AR, 3D interfaces, and physical and tangible interaction for conducting multimedia musical performance. Augmented Groove provides a collaborative environment with or without traditional music instruments, and interaction is achieved by manipulating physical cards on a table, which is mapped to changes in musical elements such as timbre, pitch, rhythm, distortion, and reverb.

Pair et al. (2002) present a real time 3D visual effects system developed for the band Duran Duran's December 2000 "Pop Trash" live concert tour which provided band members the apparent ability to pick up animated characters and interact directly with them on stage.

Oliver and Pickles (2007) introduce Fijuu, a 3D audio-visual performance environment built with the open-source game engine Ogre, in which the user can record loops and manipulate sound parameters like amplitude and rotation speed, using a PlayStation gamepad controller.

Hamilton (2008) introduces Q3osc as a real-time networked performance and spatialization environment, built with a modified version of the open-sourced ioquake3 game engine, that features sonification of in-game object actions, such as projectiles bounces, through OSC communication between the game server and audio servers.

Berthaut, Desainte-Catherine & Hachet (2011) focus on immersive virtual environments, and present the 3D reactive widgets that enable efficient and simultaneous control and visualization of musical processes, and introduce Piivert, a device designed for the manipulation of these widgets, through 3D musical interaction techniques.

C. Musical Motion Metaphors

As explained in Aristotle's 'Poetics', the metaphor brings a shift of meaning, given its analogy characteristics, i.e., it creates new links between different conceptual domains. Metaphor has had a prominent role in realm of Western philosophy and in literature, for stylistic purposes.

Lakoff and Johnson (1980) highlight their importance as a fundamental device in cognitive processes. They argue that metaphors are pervasive in everyday life, in language, thought and action, supporting our understanding of abstract concepts such as time, state, change, causation, action, purpose, etc. The use of metaphors extends beyond nonlinguistic domains as they can be expressed in different modalities.

Spitzer (2004) addresses the role of metaphor in the conceptualization of music in the processes of its reception and production: "With reception, theorists and listeners conceptualize musical structure by metaphorically mapping from physical bodily experience. With production, the illusion of a musical body emerges through compositional poetics".

Johnson and Larson (2003) claim that our understanding of musical motion is entirely metaphoric, and that these key metaphors are grounded in bodily experiences of physical motion. Experiences such as seeing objects move, moving our bodies and feeling our bodies being moved by forces, confer internal logic to such metaphors. Authors present a set of three musical motion metaphors based on the following inferences drawn from physical motion: (a) motion requires an object that moves, (b) motion takes place along a path, and (c) motion will have a manner. These metaphors, the Moving Music (Table I), the Musical Landscape (Table II) and the Music as Moving Force (Table III) are based on a set of complex mappings that combine a notion of physical motion with the metaphorical entailments of the "MOVING TIMES" metaphor. In this metaphor, temporal change is understood as a particular kind of motion through space, where times are conceptualized as objects moving toward and then past the stationary observer (Johnson and Larson, 2003).

TABLE 1. THE MOVING MUSIC METAPHOR, ADAPTED FROM JOHNSON AND LARSON (2003)

Physical Motion (Source)	Music (Target)
Physical object	Musical event
Physical motion	Musical motion
Speed of motion	Tempo
Location of observer	Present musical event
Objects in front of observer	Future musical events
Objects behind observer	Past musical events
Path of motion	Musical passage
Starting/ending point of motion	Beginning/end of passage
Temporary cessation of motion	Rest, caesura
Motion over same path again	Recapitulation, repeat
Physical forces (e.g., inertia, gravity, magnetism)	"Musical forces"

TABLE II. THE MUSICAL LANDSCAPE METAPHOR, ADAPTED FROM JOHNSON AND LARSON (2003)

Physical Space (Source)	Musical Space (Target)
Traveler	Listener
Path traversed	Musical work
Traveler's present location	Present musical event
Path already traveled	Music already heard
Path in front of traveler	Music not yet heard
Segments of the path	Elements of musical form
Speed of traveler's motion	Tempo

TABLE III. MUSIC AS MOVING FORCE METAPHOR, ADAPTED FROM JOHNSON AND LARSON (2003)

Physical Motion (Source)	Musical Experience (Target)
Locations	Emotional States
Movement (from place to place)	Change of emotional state
Physical forces	Causes
Forced movement	Causation
Intensity of force	Intensity of musical impact

III. DESIGN CONSTRAINTS AND DECISIONS

For this project, a set of design constraints was established up ahead. Unity3D game engine [23] was set to be used as the main development platform and runtime environment for exploration and prototype production.

Unity3D is a development environment specifically tailored for video games production and very efficient for building working prototypes. It allows the integration of 3rd party plugins and assets that are available at the official Unity3D asset store, which offers some of the perks of having community involvement. A great advantage of Unity3D is the set of target platform deployments it supports. It enables the development of applications that can target the range of the most representative commercial devices and operating systems. This feature is instrumental for the project, as we are developing prototypes for both mobile and standalone desktop applications.

However, Unity3D is somewhat lacking in sound capabilities for our specific needs. Its internal FMOD[2] powered audio engine has audio playback, looping, and spatialization features that provide great support for most typical use-cases for sound design in game development, although its lacks capabilities in analysis, synthesis and procedural audio of many of the available alternatives platforms for audio development.

This raised the need to look for and into more adequate technologies that would provide the flexibility we sought for in an audio engine, but that would also integrate with Unity3D and observe further requirements. For that, we performed a weighted analysis of the existing possibilities, such as Marsyas, Max/Msp, Supercollider, and PureData (Pd) regarding a set of features that align with our objectives and project requirements. From these alternatives, we have chosen Pd as the platform to build our audio engine.

Pd is an open-source data flow programming language developed by Miller Puckette at Ircam, and was originally intended for composers and visual artists. Besides offering the possibility of dynamic patching for real-time construction of audio analysis and synthesis devices, it is flexible in terms of deployment, namely for mobile devices. Furthermore, Pd has the support of a rather active community and enables the integration of open-source components, a feature that shall be further explored in the paper. Unfortunately, Pd lacks a multi-instantiation mechanism, which is taken for granted in any object-oriented language, in Supercollider, and in Max/Msp, through the Poly~ object. Nevertheless, in our perspective, the sum of the other features makes up for this disadvantage.

Another requirement was the integration of 3D motion capture technologies with Unity3D for multimodal gestural acquisition. We have used 3D sensors such as Leap Motion and Kinect, and a Vicon motion capture system as supporting technologies for real-time capture of gestural performance.

IV. IMPLEMENTATION

This section provides an overview of the design and proposed implementation, covering the general architecture, the assumptions and decisions concerning the interface,

[2] FMOD (Firelight Technologies Pty, Ltd., 2011)

control space, sound space, the mapping strategies and feedback.

A. General Architecture

The system consists of a common application core for SAR and a mobile environment that comprises an audio engine component and a sensing layer component that seamlessly integrates a set of sensors according to the specifics of each setting. Figure 1 presents the architecture of the system.

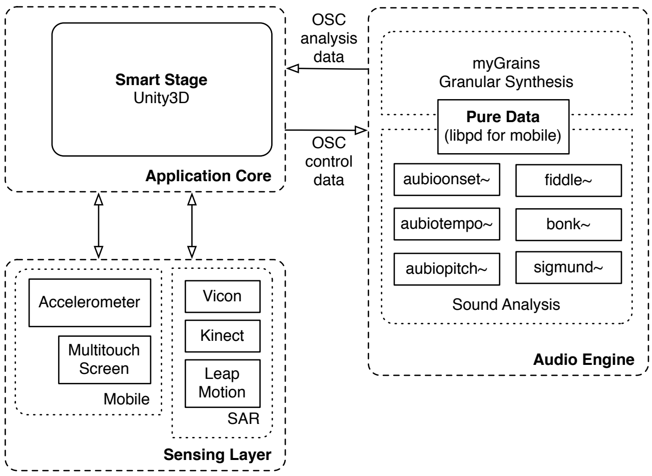

Fig 1. Smart Stage general architecture

The application core is developed in the Unity3D game engine and provides an interface with a visual metaphor and narrative built upon its three-dimensional physics engine, with real-time audio analysis visualizations, multi-parametric sound control and feedback.

Depending on the deployment settings, the application core will receive specific control events from sensors bounded to the application context. In SAR settings, the application uses control input from 3D motion capture sensors such as Leap Motion (n.d.), Kinect [25] and Vicon T-Series motion capture system (Vicon, n.d.), if deployed and active within the infrastructure environment. The SAR visual interface is projected into a surface using video projectors and projection mapping techniques. The sensing layer is integrated in the application core using specific Unity3D assets that have been either downloaded from the asset store and modified, or developed purposely for our needs, such the OSC (Wright & Freed, 1997) client implementation for communication with the Vicon system. Thus, in the SAR setting, the user will be allowed to perform music through body and gestural interaction.

In mobile settings, the control input comes from the mobile device built-in sensors, such as the three-axis accelerometer and the multi-touch screen. The user will perform music by tilting the mobile device and through touch gestures with the device's screen. This is a subset of the intended interaction capabilities for mobile, as we intend to develop further affordances using the camera, the gyroscope, the magnetometer and the microphone, and also communication and service consumption with the infrastructure.

We use libpd, a compact and acknowledgedly stable wrapper that turns Pd into an embeddable audio library [28], with specific bindings for our Unity3D application core. The design here proposed is similar to what has been adopted in some acknowledged projects in game development, such as EA's Spore, Darkspore and Dead Space [29], FRACT (Phosfiend Systems, 2014) mobile music apps such as RjDj (Reality Jockey Ltd., 2013), and so many research projects in digital music interfaces (DMI) research.

B. A First Case Study of a 3D Interaction Metaphor in Smart Stage

The three metaphors suggested are not compatible among each other, which is acknowledged by Johnson and Larson (2003). And some of their entailments are more easily implemented than others. We address several of the key metaphors and entailments involved in the conceptualization of the interface, following the original notation of the authors, by using small capital letters, and grouping them according implementation ease and success:

Metaphor: TRAVELER IS A LISTENER
Metaphor: PATH TRAVERSED IS A MUSICAL WORK

These metaphors have a straightforward implementation within our system. While experiencing the system, the user is in fact a listener, to which we would further extend the role to performer, given the active control he exerts in the audiovisual outcome. The same occurs with the traversal of the path, given that the audiovisual narrative has been created for the effect. In our system, this metaphor also cares for extension, from musical work to audiovisual work.

Metaphor: A PHYSICAL OBJECT IS A MUSICAL EVENT
Metaphor: PHYSICAL FORCES ARE MUSICAL FORCES

Each object in the interface is a sound agent, i.e., it has a sonic impact in the experience. Interactions between elements follow a natural behavior based on a physics engine upon which they are implemented. Forces applied to elements, or between them such as attraction and collisions, friction between surfaces, swarm behaviors have a sonic impact or cause sound effects.

Metaphor: SEGMENTS OF THE PATH ARE ELEMENTS OF MUSICAL FORM

For Johnson and Larson (2003), musicians often perspective the analysis of a score as a metaphorical representation, an imaginary path through an abstract musical space. This metaphor would be easily achievable with a visualization based on MIDI formats, with and not as much with audio analysis, whereas resulting data is not clear. In either case it could portray an interesting aesthetic choice, and perhaps a functional one by contributing to the enlightenment of the user. This would imply to carefully and coherently define metaphor entailments such as mappings to notes, rests, rhythm and the all the other elements of music notation. This could possibly defy the minimalist aesthetics of the interface, but be a case of further analysis to be relegated to future developments.

Metaphor: PHYSICAL MOTION IS MUSICAL MOTION
Metaphor: SPEED OF MOTION IS TEMPO
Metaphor: SPEED OF TRAVELER'S MOTION IS TEMPO

The first metaphor is general enough to be considered successfully implemented. The last two metaphors are in opposition though, and decisions have to be taken regarding their application. The second metaphor indicates that the general speed of motion of interface elements must be determined by the music tempo. In this case, the outcome of the tempo detection algorithm is mapped to the speed of the tunnel traversal. In the third metaphor, not only the visual elements but also music reproduction is dependent of the motion of the performer which implies taking resort of granular synthesis for time stretching of the musical outcome. In the current state of implementation, we have compromised the implementation in by subordinating the traversal to music tempo and some of the visual elements and music events to the performer's motion capture data.

Metaphor: TRAVELER'S PRESENT LOCATION IS PRESENT MUSICAL EVENTS

Metaphor: LOCATION OF OBSERVER IS A PRESENT MUSICAL EVENT

In these metaphors there is a convergence and possibly overlapping between the roles of traveler and observer. Nevertheless, musical events are visualized in the origin of the z-axis of the coordinate system in our environment, which corresponds to the projection plane. The actions of the performer have a direct visual impact in that point.

Metaphor: OBJECTS IN FRONT OF OBSERVER ARE FUTURE MUSICAL EVENTS

Metaphor: PATH IN FRONT OF TRAVELER IS MUSIC NOT YET HEARD

These metaphors have been implemented by making silent the upcoming elements of visual traversal, until reaching the origin, causing a sonic impact at that point. Nonetheless, a more successful approach would be to follow a generative approach, and perform a previous analysis of pre-recorded input to predefine paths and elements of the traversal.

Metaphor: PATH ALREADY TRAVELED IS MUSIC ALREADY HEARD

Metaphor: OBJECTS BEHIND OBSERVER ARE PAST MUSICAL EVENTS

The interface is built with a viewport starting at origin of the z-axis until infinite. Elements passing the origin of the z-axis no longer exist in the viewport.

C. *Graphical User Interface and Audiovisual Narrative*

Our approach to interaction design attempts to map the conceptual musical motion metaphors to multimodal gestural acquisition and 3D graphics rendered for video projection mapping. The graphical user interface presents a 3D scene in which visual elements, hierarchically grouped 3D primitives such as spheres, a cylinder primitive and flat circles, are rendered in wireframe mode, with minimal aesthetics, in black and white. It has an adaptive behavior in which the narrative is triggered by an opt-in routine based on the detection of a user within the Kinect sensor range. It begins with rendering of the central large sphere and the playback of an introductory audio sample. If the user's hands enter the range of the Leap Motion, a new sample is triggered, and the central large sphere begins a pulsating movement, scaling on a factor dependent of FFT analysis of that sample. Furthermore, little spheres are rendered as satellites of the central sphere, influenced by the dynamics of the user's fingers movements. Collisions of the satellite spheres with the central one may succeed, and will trigger sounds.

If motion dynamics reach a certain threshold, the cylinder is rendered longitudinally, and the group of spheres is placed at the extremity of a tunnel, as about to traverse it. In a glimpse, a new sample starts, and the traversal begins. As the group of spheres travels through the tunnel, they swirl around and collision with the tunnel's walls may succeed, with the consequent generation and modification of sounds. The cylinder walls become distorted, with the mesh distortion reflecting the analysis and dynamics of sound input.

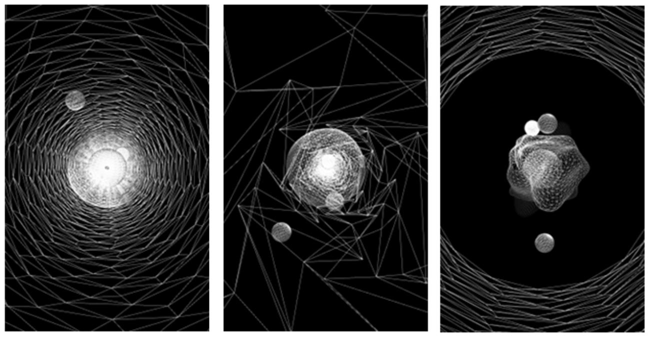

Fig 2. Evolution of the narrative in Smart Stage's GUI

The user must control the group of spheres throughout the traversal until it exits the cylinder. If the user leaves the range of detection, the scene is progressively deconstructed to the initial state. Figure 4 shows beginning, middle and end of traversal.

D. *Control Space and Gestures*

We aim for a system design that enables us to achieve a meaningful compromise between influence and control, between novice and instrumental affordances. 3D motion capture provides a very meaningful way to delve into this problem and to explore multidimensional control, as it enables six degrees of freedom (DOF) interaction and support for the simultaneous manipulation of multiple and interdependent parameters. By using three different kinds of 3D motion capture technologies, Leap Motion, Kinect and Vicon motion capture, we need to deal with real-time sensor data fusion, redundancy, and overlapping and inaccurate information.

In order to do so, we use strategies and mechanisms that have inspired us in research around context-aware applications [32]. Our sensing layer is designed to acquire positional, body and gestural interaction data, in the SAR setting, and tilt and touch gestural data in the mobile setting. Thus, we are acquiring the situational context (Salber, Dey & Abowd, 1999) of the SAR space and of the mobile device, focusing on user interaction specifics. Inspired on the works of Crowley et al. (2002) and Coutaz & Rey (2002) we have designed our sensing layer as a federation of contextors (Crowley et al., 2002), distributed along different abstraction levels, that acquire sensor data to transform in context data for higher level application consumption. For instance, a low-level sensing contextor is attached to Leap Motion controller, which delivers hand gesture data to a mid-level contextor, which in turn performs data fusion with Kinect hands articulation data, originating from another Kinect low-level contextor.

Our resulting control space is thus defined as the sum of all the ranges of possibilities given by each motion capture sensor. For instance, Leap Motion sensor provides an application-programming interface (API) that provides data regarding the identification of digits, position and orientation, and extendedness of each finger bone for the five fingers for each hand, right or left-handedness, pinch or grasp detection, etc.

Figure 3 a) and b) depicts a SAR installation of Smart Stage prototype with sensors by the author.

Fig 3 a) and b). Deployment space and real installation

Videos of the installation of an initial prototype are available (Bernardo, 2014a; Bernardo 2014b).

E. Sound Space and Audio Engine

The Pd-based audio engine provides two main functions: analysis and synthesis. We have designed a Pd patch that encompasses sub-patches dedicated to each major function. For the analysis, we extract as many features as possible from incoming sound in order to inform visualization and feedback in the application layer. To do that, we use several Pd objects, such as Fiddle, Bonk and Sigmund (internal Pd objects that are default to any Pd distribution) and Aubio [36], a set of open source external object that we have ported to Pd, in order to test and compare accuracy and performance.

- fiddle~ – outputs the pitch and amplitude of an incoming sound, and a list of sinusoidal peaks used to make the pitch determination.
- bonk~ - detects transient attacks, as sharp changes in the spectral envelope of the incoming sound, resulting from percussive sounds.
- sigmund~ - analyzes an incoming sound into sinusoidal components, which may be reported individually or combined to form a pitch estimate.
- aubioonset~ - like bonk~ object, it detects transient attacks (onsets) plus non-percussive attacks.
- aubiotempo~ - detects beat locations.
- aubiopitch~ - attempts to label each frame of the input sound with a pitch.

For this primary design version we begin with a specific functionality, which is granular synthesis applied to sound input. The granular synthesis method is out of the scope of this paper but in brief, it is a synthesis method whereby a sound particle (a sound sample of 1 to 50ms or more) is imitated, magnified, and layered with multiple imitation particles that are either cloned or extracted through a similar process as the original to create different sounds (Curtis Roads, 1996, Truax, 1988). Our granular synthesizer, myGrains, is a modified version of Grains, a Pd patch developed by Tim Vets (n.d.). It exposes the following control parameters:

- Grain speed – automated sample rate factor;
- Grain pan spread - simultaneous sample spatialization factor;
- Grain pitch spread – simultaneous sample transposition factor;
- Grain multiplication – amount of samples used in synthesis;

F. Mapping strategies and feedback

Inspired by the enactive approach to instrument design by Essl and O'Modhrain (2006), we decided to take advantage of situations where VO interactions appear to have a natural behavior, such as attraction and collisions, friction between surfaces, swarm behaviors, etc. From the amount of possibilities within our Smart Stage environment, and in an intuitive fashion to test our proof-of-concept, we have decided for the set presented in Table 4.

TABLE IV: PROPOSED MAPPINGS FOR CONTROL SPACE - SOUND SPACE

Leap Motion	Finger spatial position Hand distance to sensor In-between hand distance	Satellite VO movement Central VO navigation Central VO scale
Kinect	Silhouette User's relative position to LeapMotion sensor User's abs. position	Direct Feedback Parallax/Perspective Opt-in/out trigger Parallax/Perspective
Vicon MoCap	User's rel. position to LeapMotion sensor	Parallax/Perspective Opt-in/out trigger
Mobile	XY multitouch Tilt Pinch	Satellite VO movement Central VO navigation Central VO scale
Sound Input	Tempo Transient, Spectral Analysis	Grain speed VO mesh distortion
VO	Collisions Friction Satellite VO distance to Central VO	Percussive sounds Grain multiplication Grain pitch spread Grain pan spread

According to Jordà (2005), "'audio' feedback presented to the performer in an interactive visual form, intuitively helps the understanding and the mastery of the interface, enabling the simultaneous control of a high number of parameters that could not be possible without this visual feedback" (Jordà, 2005, p.396). We share this perspective in what concerns the integration of audio representation with control. The VOs and composites within the 3D environment scene provide audiovisual feedback through mesh distortion processes, as they are both visualizations of real-time audio analysis data and sources of multi-parametric sound control. In addition, like Mulder and Fels (1998) we imprint intuitive feedback representations in the VOs, that result the from the gestural expression data, to inform the user's perceptual system about the control space, so that he comprehends intuitively and exerts a faster and intuitive control.

V. Conclusion and Future Work

The Smart Stage system is a work in progress, in an early stage of conception and development, given that it stands at the intersection of several areas of research and that its underlying objectives are ambitious. The main effort in this paper was to lay the foundations and document the decisions behind the design process.

For the scope of this article, we tried to limit our approach as much as we could and present a high level, end-to-end perspective of Smart Stage, focusing on the technological assumptions that support our vision, and providing a case study with a minimal set of features that would allow us to instance and test the concept and some of its major components.

We reviewed the literature around UbiComp, 3D music interaction and immersive environments and metaphors in music in order to situate our research and to inform the design process and presented a technical overview of the system that covered decisions regarding architecture, interface, control space, sound space and mappings. We also presented a first case study of a 3D musical interaction metaphor for granular synthesis developed for this environment.

The architecture has been designed according to initial constraints. Several other alternatives could be considered also, with intrinsic benefits and disadvantages certainly. In fact, in the course of this effort we discovered several constraints or impediments that may lead to refactoring. For instance, Unity3D does not provide much flexibility in what concerns its runtime configuration and software architecture; also, there are open source platforms from which we could take better advantage of the community effort; PureData does not provide multi-instantiation, Kinect has a much more limited API in Mac OS than on Windows. These are some examples of the issues that have to be considered.

For future work we intend to address the several of the previously stated features such the integration between mobile applications and SAR infrastructure, infrastructural services such as SMPTE sync and shared control, collaboration and support for different roles. In order to proceed in this direction, an in-depth HCI study has to be performed in order to characterize user roles, goals, and tasks and define use cases scenarios, and to inform a comprehensive evaluation. In the mobile setting, we also aim to take advantage of the available full range of mobile device's sensors. In the audio engine we intend to develop AI-based compositional generators and support for more synthesis techniques.

References

[1] J. Gibson, "The ecological approach to the visual perception of pictures," *Leonardo*, vol. 11, no. 3, p. 227, Jan. 1978.

[2] M. L. Johnson and S. Larson, "'Something in the Way She Moves' - Metaphors of Musical Motion," *Metaphor Symb.*, vol. 18, no. 2, pp. 63–84, 2003.

[3] M. Weiser, "The Computer for the 21st Century," *Sci. Am.*, vol. 265, no. 3, pp. 66–75, Jan. 1991.

[4] J. C. Augusto, H. Nakashima, and H. Aghajan, "Ambient Intelligence and Smart Environments: A State of the Art," *Handb. Ambient Intell. Smart Environ.*, pp. 3–31, 2010.

[5] Y. Rogers, "Moving on from weiser's vision of calm computing: Engaging ubicomp experiences," *UbiComp 2006*, pp. 404–421, 2006.

[6] M. S. Pimenta, L. V Flores, A. Capasso, P. Tinajero, and D. Keller, "Ubiquitous music: Concepts and metaphors," in *Proceedings of the XII Brazilian Symposium on Computer Music*, 2009, pp. 139–150.

[7] A. Tanaka, "Mobile music making," *Proc. 2004 Conf. New interfaces …*, pp. 5–7, 2004.

[8] G. Essl and M. Rohs, "Sound: Interactivity for Mobile Music-Making," no. June, pp. 197–207, 2009.

[9] L. Flores, M. Pimenta, E. R. Miranda, E. A. A. Radanovitsch, and D. Keller, "Patterns for the design of musical interaction with everyday mobile devices," *Proc. IX …*, vol. 5, no. 8, 2010.

[10] A. Barbosa, "Displaced Soundscapes: A Survey of Network Systems for Music and Sonic Art Creation," *Leonardo Music J.*, vol. 13, pp. 53–59, 2003.

[11] T. Blaine and S. Fels, "Collaborative Musical Experiences for Novices," *J. New Music Res.*, vol. 32, no. 4, pp. 411–428, Dec. 2003.

[12] M. Kaltenbrunner, S. Jorda, G. Geiger, and M. Alonso, "The reacTable*: A Collaborative Musical Instrument," *15th IEEE Int. Work. Enabling Technol. Infrastruct. Collab. Enterp.*, pp. 406–411, 2006.

[13] F. Morreale, A. De Angeli, R. Masu, P. Rota, and N. Conci, "Collaborative creativity: The Music Room," *Pers. Ubiquitous Comput.*, vol. 18, no. 5, pp. 1187–1199, Oct. 2013.

[14] M. M. Wanderley and N. Orio, "Evaluation of Input Devices for Musical Expression: Borrowing Tools

from HCI," *Comput. Music J.*, vol. 26, no. 3, pp. 62–76, Sep. 2002.

[15] P. Vajpeyi, "Designing Rich Sensory Experiences with Strategies of Transformation and Augmentation," New School University, New York, 2001.

[16] G. Levin, "Painterly Interfaces for Audiovisual Performance," Massachusetts Institute of Technology, 2000.

[17] A. Mulder and S. Fels, "Sound Sculpting: Performing with Virtual Musical Instruments," ... *fifth Brazilian Symp. Comput. Music. …*, no. August 1998, pp. 151–164, 1998.

[18] I. Poupyrev, R. Berry, and J. Kurumisawa, "Augmented Groove: Collaborative Jamming in Augmented Reality," in *SIGGRAPH 2000 Emerging Technologies*, 2000, p. 19.

[19] J. Pair, J. Wilson, J. Chastine, and M. Gandy, "The Duran Duran project: the augmented reality toolkit in live performance," in *The First IEEE International Workshop Agumented Reality Toolkit,*, 2002, p. 2.

[20] J. Oliver and S. Pickles, "Fijuu2: a game-based audio-visual performance and composition engine," *Proc. 7th Int. Conf. New Interfaces Music. Expr.*, 2007.

[21] R. Hamilton, "Q3OSC OR: HOW I LEARNED TO STOP WORRYING AND LOVE THE BOMB GAME," *Proc. Int. Comput. Music …*, 2008.

[22] F. Berthaut, M. Desainte-Catherine, and M. Hachet, "Interacting with 3D Reactive Widgets for Musical Performance," *J. New Music Res.*, vol. 40, no. 3, pp. 253–263, Sep. 2011.

[23] Unity Technologies, "Unity3d." [Online]. Available: http://unity3d.com. [Accessed: 04-Jul-2014].

[24] L. Motion, "Leap Motion," 2014. [Online]. Available: https://www.leapmotion.com. [Accessed: 03-Jul-2014].

[25] Microsoft Corporation, "Kinect for Windows | Human Interface Guidelines v1.8," 2013.

[26] V. M. S. Ltd, "Vicon T-Series." [Online]. Available: http://www.vicon.com/System/TSeries. [Accessed: 03-Jul-2014].

[27] M. Wright and A. Freed, "Open sound control: A new protocol for communicating with sound synthesizers," in *Proceedings of the 1997 International Computer Music Conference*, 1997, pp. 101–104.

[28] P. Brinkmann, P. Kirn, R. Lawler, C. McCormick, M. Roth, and H.-C. Steiner, "Embedding pure data with libpd," in *PDCON:Conference*, 2011, p. 8.

[29] K. Jolly, "Usage of Pd in Spore and Darkspore," in *PureData Convention*, 2011.

[30] P. Systems, "FRACT OSC," 2014. [Online]. Available: http://fractgame.com. [Accessed: 03-Jul-2014].

[31] R. J. Ltd., "RjDj," 2013. [Online]. Available: http://rjdj.me. [Accessed: 03-Jul-2014].

[32] B. Schilit and M. Theimer, "Disseminating Active Map Information to Mobile Hosts," *Network, IEEE*, 1994.

[33] D. Salber, A. Dey, and G. Abowd, "The context toolkit: aiding the development of context-enabled applications," *... SIGCHI Conf. Hum. …*, 1999.

[34] J. Crowley, J. Coutaz, G. Rey, and P. Reignier, "Perceptual components for context aware computing," *... 2002 Ubiquitous Comput.*, 2002.

[35] J. Coutaz and G. Rey, "Foundations for a Theory of Contextors," *Comput. Des. User Interfaces III*, 2002.

[36] P. Brossier, "Aubio," 2009. [Online]. Available: http://aubio.org/. [Accessed: 03-Jul-2014].

[37] C. Roads, *The Computer Music Tutorial*, The MIT Pr. Cambridge and London, 1996.

[38] T. Vets, "Another 'classic': Granular Synthesis in Pd," *http://www.timvets.net*. [Online]. Available: http://www.timvets.net/video/grains.php?page=software. [Accessed: 03-Jul-2014].

[39] G. Essl and S. O'Modhrain, "An enactive approach to the design of new tangible musical instruments," *Organised Sound*, vol. 11, no. 03, p. 285, Nov. 2006.

[40] A. van Dam, "Post-WIMP user interfaces," *Commun. ACM*, vol. 40, no. 2, pp. 63–67, Feb. 1997.

Francisco Bernardo, Eng, MSc, M.A., is Invited Assistant Professor and PhD student at Catholic University of Portugal (UCP - http://artes.ucp.pt/), and researcher at Research Center for Science and Technology in the Arts (CITAR) in VisualYzARt project. His research is focused on human computer interaction, ubiquitous computing, augmented reality, natural user interfaces, music technology, audiovisual performance and aesthetics, software development, music tech, music business and creative industries. He holds a graduate degree in Computer Science and Systems Engineering since 2006, and master degree in Mobile Systems since 2009, both from University of Minho. In 2013, he finished his second master degree in Management of Creative Industries, from Catholic University of Portugal. His research interests are in. He also has over seven years of professional experience in the software industry as a developer.

Pedro Duarte Pestana, PhD, is an enthusiast of all things sound related. He has worked as a sound engineer for 20 years, having recorded and mixed several high-profile artists in his native Portugal. He has taught at several universities and is currently in charge of music production in the School of Arts of the Catholic University of Portugal. His research interests include DSP and engineering techniques, having recently finished a PhD on computer assisted mixing systems. He joined the Canadian company Mixgenius in 2013, where he consults on intelligent systems, engineering and signal processing.

Luís Gustavo Martins, PhD, is Assistant Professor at the Sound and Image Department of the School of the Arts of the Catholic University of Portugal (UCP - http://artes.ucp.pt/), and a researcher at the Research Center for Science and Technology in the Arts (CITAR), Porto, Portugal and responsible researcher for VisualYzARt project. His research is mainly focused on audio content analysis, sound processing and synthesis, and his research interests include signal processing, machine learning, perception and cognition, music information retrieval (MIR) and software development. More recently he has been exploring the use of tangible and multitouch interfaces for sound and music exploration and interaction. He is the Principal Investigator (PI) of the Research Project "A Computational Framework for Sound Segregation in Music Signals", funded by the Portuguese Foundation for the Science and Technology (FCT) and with reference PTDC/EIA-CCO/111050/2009 (https://sites.google.com/site/casafctproject/home). He is an active developer of the open source Marsyas audio processing software framework (http://marsyas.sf.net/).