Joint Virtual Reality Conference of EGVE - ICAT - EuroVR (2009)
M. Hirose, D. Schmalstieg, C. A. Wingrave, and K. Nishimura (Editors)

Opportunistic Music

M. Hachet[1], A. Kian[2], F. Berthaut[2], JS. Franco[2], and M. Desainte-Catherine[2]

LaBRI - [1] INRIA Bordeaux, [2] Université de Bordeaux

Abstract

While mixed reality has inspired the development of many new musical instruments, few approaches explore the potential of mobile setups. We present a new musical interaction concept, called opportunistic music. *It allows musicians to recreate a hardware musical controller using any objects of their immediate environment. This approach benefits from the physical attributes of real objects for controlling music. Our prototype is based on a stereo-vision tracking system associated with FSR sensors. It allows musicians to define and to interact with opportunistic tangible widgets. Linking these widgets with sound processes allows the interactive creation of musical pieces, where musicians get inspiration from the surrounding environment.*

Categories and Subject Descriptors (according to ACM CCS):
H.5.1 [Information interfaces and presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities
J.5 [Arts and humanities]: Music, Arts, fine and performing—

1. Introduction

Musicians interact with musical applications by the way of controllers. These controllers are either based on physical devices such as mixing tables or control pads, or they lie on graphical interfaces. Physical devices allow direct and efficient control of the musical parameters. For example, a physical fader is commonly used to adjust precisely the volume of an audio track. On the other hand, the structure of the physical devices as well as the number of their functionalities is fixed. On the contrary, graphical interfaces are not limited by physical constraints and therefore offer infinite configurations. For example, a virtual fader controlled by way of a mouse can be displayed using various graphical representations. On the other hand, these interfaces suffer from a lack of haptic feedback, which decreases the efficiency of the control.

We propose a new approach called *opportunistic music* for interaction with musical pieces. The idea is to benefit from the physical attributes of the surrounding objects for controlling music. For example, a musician will slide his finger along the border of a table to control the volume of an audio track, or he will tap on the cover of a book to start a musical loop. This approach is inspired by the work of Henderson and Feiner [HF08], where opportunistic controls are used for interaction with AR applications (eg. maintenance).

Figure 1: *A phone used as an opportunistic fader to control a sound parameter.*

Such an opportunistic approach opens new perspectives for musical creation. By interacting with the physical objects

DOI: 10.2312/EGVE/JVRC09/045-051

surrounding them, the musicians benefit from inexhaustible sources of inspiration. The way they interact with the music can evolve depending on the available surrounding objects. The musicians are no longer limited to fixed setups. They can explore new controls for interaction with their musical pieces. This approach offers extensible control, while ensuring physical interaction. Therefore, it offers opportunities to artists for a new kind of musical creation.

We have developed a first prototype to explore this concept. It is based on a stereo-vision system associated with *Force Sensitive Resistors* (FSR) sensors for the tracking of the musician's gestures (see Figure 1). With this system, musicians define virtual controllers directly on the physical environments surrounding them. Then, they can connect these controllers to sound parameters. Finally, the musicians play by directly interacting with the real objects they have defined as active objects.

2. Related Work

The past decade has seen the growth of alternative musical controllers. Furthermore, many musicians and software developers have taken advantage of research done in the mixed reality field.

The most widespread of these new controllers are *tangible musical tables*, which take inspiration from Fitzmaurice's graspable user interfaces [FBB95]. In these applications, users manipulate real objects whose 2D position and orientation are associated to parameters of sound processes. A simple example of this category is d-touch [CSR03], which allows musicians to control for example a sequencer by arranging real objects representing sounds on a grid layed on a table. Other examples include The Music Table [BMHS03], Xenakis [BCL*08] or Scrapple [Lev06]. The most advanced applications are the Audiopad [PRI02] and the Reactable [JKGB05], which both provide advanced musical control and rich visual feedback. However, these controllers rely on objects with markers and on fixed setups, i.e. large tables with fixed cameras and projectors.

Other controllers allow for more flexible setups. For example, the concept of *accoustic tangible interfaces* [CP05] gives the possibility of turning most surfaces and objects into accurate multitouch musical controllers, but it requires to equip them with calibrated audio sensors and thus is not mobile.

The "Augmented Groove" [PBB*01] application relies on 3D manipulation of real LP records which are equipped with fiducial markers. Each of these records is associated with a musical loop, which is started when the user flips the record to reveal the marker. The 3D movements then control audio effects applied on the loop, such as pitch, distortion, filter, volume. It could easily be turned into a mobile setup, like the project developped by Goudeseune [GK01]. This augmented reality application allows users to create virtual sound sources in a real environment. Musicians can then modify the sources' positions and navigate through them.

The "Sound of Touch" [MR07] is another interesting instrument. Users record sounds of the environment with a wand and play them by scratching or hitting the wand against real objects. Physical properties, i.e. resonance or texture, of the real objects alter the sound properties, making it softer or stronger. Thus this instrument allows users to explore surrounding objects, and it takes advantage of the haptic feedback from the physical textures. However, musical interaction is limited to triggering the recorded sounds.

None of these musical applications combine mobility, use of unprepared objects, haptic feedback and unrestricted control possibilities. That is why we believe that *opportunistic music* will bring completely new musical possibilities to musicians.

3. Opportunistic Widgets

The key idea of opportunistic music is to interact directly with the real objects surrounding the user. We call these interactive objects *opportunistic widgets*. Numerous objects of the real environment may have interesting properties when used as opportunistic widgets. We propose to take advantage of these physical characteristics to enable tangible interaction. It has been shown that tangible interaction may improve the performance of users for the completion of interactive tasks. In the case of music, we are convinced that such an approach, which provides sensitive feedback to the musician, can be very valuable. For example, furniture edges are well suited to control linear values. By sliding their hands along the edge of the furniture, the musician interact as they would do with a standard fader. Another example is the use of physical objects as pads. By tapping on the objects, the musicians can start events.

In addition to standard widgets such as pads or faders, the real environment may inspire new unconventional widgets that can have interesting properties for music. For example, a curved surface may produce valuable tangible feedback to the musician. Other examples are opportunistic widgets for which elastic feedback is intrinsically provided (e.g. a folder with rubber bands). For such widgets, the musicians benefit from an elastic sensing mode, which can be particularly interesting for the control of some sound parameters (e.g. physical modelling synthesis). Other physical properties of the real environment can be exploited. For example, musicians can exploit the texture of the objects, their viscosity or even their warmth as a haptic feedback. Figure 2 illustrates some examples of opportunistic widgets.

While facing the real environment, musicians can play with the objects without moving them, or they can rearrange their playing environment. For example, they can use some sheets of paper, on which they draw some relevant signs or

Figure 2: *Examples of opportunistic widgets. (left) Curved and linear widgets are defined on a desktop environment. The rubber band of the folder is used to benefit from elastic feedback. The staples box slides along the book. A post-it with an annotation is also used. (right) Outside environments may be very inspiring, as well. In this example, different natural textures are used to distinguish the widgets.*

text. Then, they can arrange these widgets as desired, before assigning them a function.

In the following, we describe the technical setup we have developed to define the opportunistic widgets and to interact with them. This concerns the low level tracking issues, the creation of the widgets, and their mapping to musical events. Our current implementation does not allow mobile use yet. In this paper, we present a first prototype that validates our concept. It is the first step towards a full mobile system.

4. Technical Environment

To build an opportunistic music system, we need to track 3D user movement and respond to user-triggered events. The system must also have desirable characteristics such as portability, on-site robustness, minimal user space cluttering, and high reactivity. The latter is a requirement for interactive systems in general, but is especially relevant to music, where typical response times cannot exceed 20ms. Vision-based tracking systems offer benefits toward portability and being minimally invasive, as they can provide the ability to rapidly process visual tracking data with minimal user instrumentation in a compact package. They also have an inherent advantage since they can easily be built with cost-efficient off-the-shelf components, as the availability of video hardware soars. However, typical camera acquisition frequencies do not exceed 30Hz for low to medium-end cameras. Also the detection of surface contact from video is fundamentally ambiguous for this type of setup as scene geometry is assumed fully unknown. Vision systems are thus desirable for identifying 3D positions and regions of movement, but not precise enough in detecting high frequency events such as surface contact. For this reason a second more responsive input source is needed. This input does not need to provide positional information, but rather a way of identifying high-frequency trigger and contact events to complement the vision system.

4.1. Current Prototype

To implement the desired system characteristics, we propose an opportunistic music prototype based on inputs from two cameras and an FSR sensor. The former allows for stereovision-based 3D positioning of visually salient objects, while the latter allows to detect pressure with high precision and responsiveness (<10ms). To allow absolute 3D positioning in the scene and the creation of world-basis music widgets, we need to acquire projective characteristics of cameras and their positional information. All such parameters are typically described by a single 3x4 *projection matrix* using the common pinhole-camera model [HZ00]. The projection matrix linearly describes where a given 3D scene point projects in camera image pixel coordinates, and can then easily be used to triangulate the 3D position of a scene point from its two identified image occurrences. In our context scene points of interest to be tracked in the scene will be the user's fingertips.

For simplicity, let us for now assume the camera pair is on a fixed mount (such as a tripod) observing the user interaction space with minimal occlusion. Depending on the scene, such a configuration can typically be obtained with over-the-shoulder camera positioning. Off-the-shelf methods can then be used to calibrate the camera-pair[†], i.e. estimate each

† Typically found in computer vision libraries such as OpenCV.

Figure 3: *Opportunistic music prototype. (left) General setup with camera pair, finger mounted FSR+LED interaction device, FSR analog acquisition board. (right) Close-up of the thimble interaction device, with FSR sensor and infrared LED.*

camera's projection matrix using a known calibration object [Zha00]. To enable user finger positioning, we still need to identify the locus of fingertip projections in camera views. To allow for robust and fast identification of fingers, we opt for a target-based approach based on infrared detection. This notably has the advantage of quasi-invariance to visible light changes and appearance-specific noise and variability, enabling detection by simple and computationally inexpensive thresholding in general situations. We thus propose minimal user instrumentation using a thimble-type device (prototype shown in Fig. 3) which hosts the FSR touch sensor and a small infrared LED. The LED could equivalently be replaced by a passive IR reflector thimble and an IR source mounted with the camera pair.

4.2. Toward a Mobile Setup

We previously assumed a fixed camera mount setup, which could be transported and used on-the-field. A drawback of this setup is that it restricts the user to a fixed tracking area observable in the common field of view of the camera pair. An idea to take this system toward a fully mobile setup is to use an instrumented headset to mount the camera pair and earphones (see Fig. 4), connected to a mobile processing unit fitted in a backpack. Head movement then allows for completely flexible interaction areas, by tracking the user's fingers in his current region of attention. A fundamental difference with the previous setup is that camera absolute positions can now be allowed to vary up to a rigid transformation, which must be estimated for each frame. We thus propose to equip the headset with a positioning device, e.g. a third camera in the visible spectrum and inertial sensor. These instruments can then be used to detect the scene's salient features and head ego-motion, thus allowing for head and camera positioning in the world coordinate frame. Music widgets can then be defined appropriately in absolute 3D coordinates in arbitrarily large interaction areas. It should be noticed that the technical solution we propose for tracking the musicians' fingers is adapted to mobile conditions, as it does not require object recognition, and it is very few dependent from the lighting conditions.

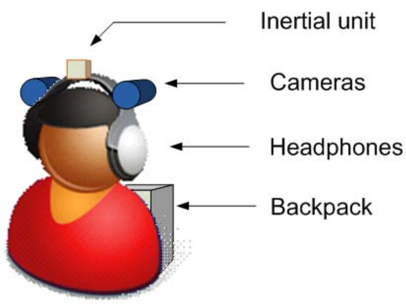

Figure 4: *Mobile setup*

5. Defining and Playing the Widgets

The system we have described in the previous section allows us to know the 3D position of the users' fingers. In addition, the FSR sensors give us precise information concerning the contact between the fingers and the physical surfaces. The next step consists in defining the opportunistic widgets.

5.1. Creating the Widgets

Widget creation should be very simple and avoid tedious procedures. To this goal, we propose registration of new widgets by simply sliding fingers on surfaces intended as widgets. The musician can then define 3D curves in a straightforward manner. These curves will be used as linear valuators (faders), or they will be used as binary inputs (pads).

Technically, when a finger pressure is detected via the FSR sensor, we record the successive 3D positions P_i of the

LED, until the user's finger is released. In addition to these 3D points, we store the local tangent vectors given by

$\vec{t_i} = \frac{P_{i+1}-P_i}{||P_{i+1}-P_i||}$

We also store the local integral distance from the origin d_i, which is given by

$d_i = d_{i-1} + ||P_{i+1} - P_i||$

At the end of the record (i.e. when the finger is released), we save an *id* as well as the bounding box given by the *min* and *max* coordinates of the curve. We also store the total integral distance d_{total}.

For each point of the curve, we can compute a corresponding normalized integral value $c \in [0,1]$, as described in the next section. These values are the output values of the opportunistic widgets.

One 3D position P_i is stored at each frame. Hence, users can define the curves accurately by sliding their fingers slowly. They can define coarse sections by moving fast, too. Hence, the users can precisely define where they need fine control. Note that in our current implementation, we use linear interpolation based on the integral distances. By taking into account the speed of the finger movements, we could use non-linear mappings, too. These non-linear mappings could be interesting to improve music expressiveness. This direction has to be studied more in depth.

Additional widgets could be defined, too. For example, 2D surfaces could be used for the bi-dimensional control of sound parameters. We will define additional widgets in our future work.

5.2. Interacting with the Widgets

The musicians can now play with the widgets they have defined. In the following, we describe how our system manages the users' input.

When a contact is detected, we first evaluate if the 3D point F corresponding to the tracked LED is in the area of an opportunistic widget. This can be done by way of a simple inclusion test with the bounding boxes of the widgets. For the concerned widgets, we estimate the distance from F to the curve, and we return the corresponding c value.

To do so, we test if F is in a tolerance cylinder for each segment $[P_iP_{i+1}]$ of the curve, as illustrated in Figure 5. We note h, the tolerance distance under which F is considered to be part of the segment $[P_iP_{i+1}]$.

We project F on (P_iP_{i+1}) and compute the scalar value

$\alpha = \vec{t}.(F - P_i)$.

F belongs to the tolerance cylinder if

$$\begin{cases} \alpha > -h \text{ and } \alpha < ||P_{i+1} - P_i|| + h \\ ||(F - P_i) - \alpha\vec{t}|| < h \end{cases}$$

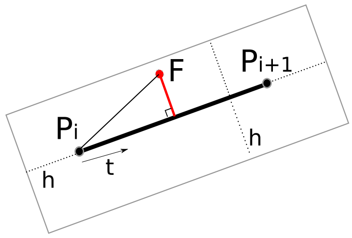

Figure 5: *Proximity test and interpolation computation*

in this case, the normalized integral value c is given by

$c = \frac{d_i + \alpha||d_{i+1} - d_i||}{d_{total}}$

This widget output value is then sent as an input to the musical subsystem. This is described in the next section.

The computation of the c values could be improved by managing the links between the segments of the curve. Indeed, our current implementation does not ensure the linear continuity of the widget output when the projection of the input points moves from one segment to another. This could be solved by linking the successive tolerance cylinders in order to ensure continuity, as illustrated in figure 6. We could also use generalized cylinders defined by parametric curves. However, in practice, the limited precision coming from the video tracking as well as the large number of recorded points do not require such an optimization.

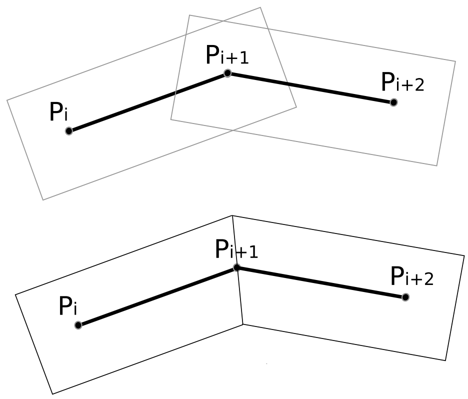

Figure 6: *Continuity of the output values. Regular cylinders (top) do not ensure full continuity. This can be improved by linking the cylinders (bottom).*

Figure 7 illustrates the output values obtained when sliding the finger along a previously defined curved-shape widget. It can be noticed that the overall continuity of the obtained values is good. The slope modifications show the speed variation of the finger movement. It can be noticed that the output curve is not perfectly smooth. This is mainly due to tracking imprecisions. To smooth the curve, we could apply a dedicated filter.

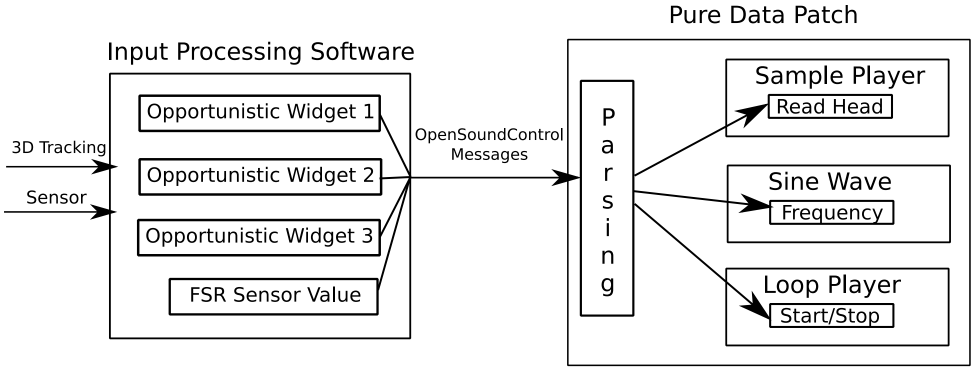

Figure 8: *An example of application with 3 opportunistic widgets associated to 3 sound parameters.*

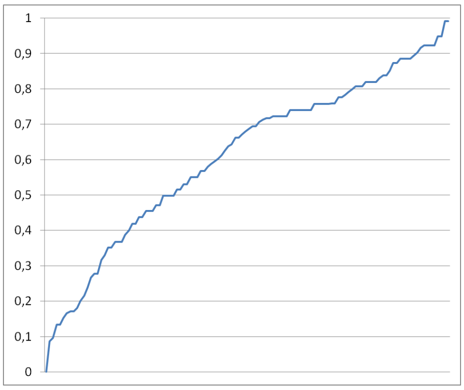

Figure 7: *Example of a widget output when sliding a tracked finger along it.*

6. Managing Sound Processes

6.1. Mapping the Widgets to the Sound Parameters

The musician needs to associate the *opportunistic widgets* to sound parameters. Numerous approaches could be used to do this, such as a voice recognition or PDA interface. In the current implementation, before creating the widgets, the musician defines physical mapping buttons by touching 3D locations in the environment. For example, for a musical application where 4 parameters can be controlled, the musician should define 4 buttons, the first one corresponding to the first parameter, the second one to the second parameter, and so on. Hence, the musician associates one physical object to one parameter (e.g. a mug is associated to loudness). The musician can also use annotated sheets of paper as labels. The mapping is done by pressing a mapping button before creating a widget as we already described. The created widget is then associated to the selected sound parameter.

6.2. OpenSoundControl Messages

The values of the opportunistic widgets and the values of the FSR sensors are sent to the audio application using the OpenSoundControl [WF97] protocol. The messages are defined as urls, widgets and FSR values being floats between 0 and 1 :

/opportun/widget1 valueWidget valueFSR

They are sent when the FSR sensor's value reaches a predefined threshold. The value of the FSR sensor is also continuously sent as */opportun/sensor valueFSR*, to enable other user-defined controls.

6.3. Sound Processes

The sound processes are actually defined as Pure Data patches, though every other musical application handling OpenSoundControl could be used. Pure Data (PD) is a visual programming language, written originally by Miller Puckette [Puc96] and aimed at developping musical and visual interactive applications. In our case, the patch parses the OpenSoundControl messages of the *opportunistic widgets*, and sends the values to the associated sound process parameter. An example of application is shown in figure 8.

6.4. Practical Example

We give here a practical example which summarizes an *opportunistic music* session from the musician point of view. We assume the musician previously defined a PD patch on his computer composed of a sine wave oscillator, a sample player, and a loop player. The user first needs to associate each sound parameter to a physical mapping button, for example the parameter's name written on a sheet of paper. Then he chooses objects of his surrounding environment (e.g. his office) he wants to play with. One possible setup may thus consist of a phone, a book, a roll of scotch-tape. The musician maps the frequency of the oscillator to the border of the phone. To do this, he starts the registration step by hitting a mapping button previously defined. Then, he slides his finger along the phone to define the widget. The widget, i.e. the 3D position composing it, is registered by the system as soon as he releases his finger. The musician can then define a new widget. For example, he can map the trigger of a drum loop to the top of the scotch tape, allowing to start/stop playing the loop when hitting its surface. He finally defines a fader controlling the read head of a sound on the border of the book, using a small box to physically keep the position of the fader. This setup allows him to start a drum loop with the scotch tape, play notes with the phone and scratch with the book. Technically, the system does not need to identify the objects by way of vision techniques. It only detects if the musician's finger belongs to a previously defined 3D area when a pressure event is detected.

7. Conclusion and Future Work

Opportunistic music is a new concept for on-site inspired musical pieces. It opens the way to new artistic creation possibilities. The first prototype we have developed allowed us to experiment this concept. Our work is too preliminary to conduct a formal user study. However, the feedback we had from artists encourage us to continue our developments in that direction.

The next step of our work will be to make the system fully mobile, as described in section 3.2. This will allow to completely leverage the concept of opportunistic music. We also plan to develop new widgets to make interaction richer, and we will experiment the use of a pico-projector to augment the musicians environments with some visual feedbacks.

Another direction we whish to explore is the integration of live-recording functionalities. By adding microphones close to the users' fingers, we could directly interact with the sounds produced by the real world. For example, the musician could interact with metallic sounds by hitting a metallic object. This live-recording concept, which as been introduced by [MR07], would be particularly interesting in our opportunistic setup.

The development of our system was motivated by musical needs. We think it could be used for other purposes, sharing the initial motivation of Henderson and Feiner [HF08]. The high reactivity we obtain thanks to the combination of FSR sensors and infrared stereo-vision could benefit to many domains. Similarly, the principle of live creation of opportunistic widgets, as well as the proposed implementation, could be valuable to other AR applications. Music inspired our work. We hope that our work will inspire new developments.

Acknowledgment

The authors would like to thank the *Studios de Création et de Recherche en Informatique et Musique Electroacoustique* (SCRIME) for the material they made available for us, and particularly Joseph Larralde for his help.

References

[BCL*08] BISCHOF M., CONRADI B., LACHENMAIER P., LINDE K., MEIER M., PÖTZL P., ANDRÉ E.: Combining tangible interaction with probability-based musical composition. In *Proceedings of the Second International Conference on Tangible and Embedded Interaction (TEI'08)* (2008).

[BMHS03] BERRY R., MAKINO M., HIKAWA N., SUZUKI M.: The augmented composer project: The music table. In *Proceedings of the Second IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2003)* (2003).

[CP05] CREVOISIER A., POLOTTI P.: Tangible acoustic interfaces and their applications for the design of new musical instruments. In *Proceedings of the 2005 International Conference on New Interfaces for Musical Expression (NIME 2005)* (2005).

[CSR03] COSTANZA E., SHELLEY S., ROBINSON J.: Introducing audio d-touch: a novel tangible user interface for music composition and performance. In *Proceedings of the 6 th international conference on digital audio effects (DAFx-03)* (2003).

[FBB95] FITZMAURICE G. W., BUXTON W., BRICKS H. I.: Laying the foundations for graspable user interfaces. In *ACM Proceedings of CHI 1995* (1995).

[GK01] GOUDESEUNE C., KACZMARSKI H.: Composing outdoor augmented-reality sound environments. In *Proceedings of the International Computer Music Conference (ICMC2001)* (2001).

[HF08] HENDERSON S. J., FEINER S.: Opportunistic controls: leveraging natural affordances as tangible user interfaces for augmented reality. In *VRST '08: Proceedings of the 2008 ACM symposium on Virtual reality software and technology* (2008), ACM, pp. 211–218.

[HZ00] HARTLEY R., ZISSERMAN A.: *Multiple View Geometry in Computer Vision*. Cambridge University Press, June 2000.

[JKGB05] JORDÀ S., KALTENBRUNNER M., GEIGER G., BENCINA R.: The reactable. In *Proceedings of the International Computer Music Conference (ICMC2005)* (2005).

[Lev06] LEVIN G.: The table is the score: An augmented-reality interface for real-time, tangible, spectrographic performance. In *Proceedings of the International Conference on Computer Music 2006 (ICMC'06)* (2006).

[MR07] MERRILL D., RAFFLE H.: The sound of touch. In *Proceedings of SIGGRAPH07* (2007).

[PBB*01] POUPYREV I., BERRY R., BILLINGHURST M., KATO H., NAKAO K., BALDWIN L., KURUMISAWA. J.: Augmented reality interface for electronic music performance. In *Proceedings of HCI.* (2001).

[PRI02] PATTEN J., RECHT B., ISHII H.: Audiopad: A tag-based interface for musical performance. In *Proceedings of Conference on New Interface for Musical Expression (NIME '02)* (2002).

[Puc96] PUCKETTE M. S.: Pure data. In *Proceedings of the International Computer Music Conference* (1996).

[WF97] WRIGHT M., FREED A.: Open sound control: A new protocol for communicating with sound synthesizers. In *Proceedings of the International Computer Music Conference* (1997).

[Zha00] ZHANG Z.: A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*, 11 (2000), 1330–1334.