Multi-touch Interface for Acousmatic Music Spatialization

Gwendal Le Vaillant
Grenoble-INP – Ense^3 (France)
gwendal.le-vaillant@ense3.grenoble-inp.fr

Rudi Giot
ISIB – LARAS (Brussels, Belgium)
giot@isib.be

ABSTRACT

The Multi-touch Interface for Acousmatic Music Spatialization (MIAM Spat) project deals with a new way of performing live music spatialization. Spatialization artists currently use hardware mixing consoles to produce three-dimensional sound effects, within concert halls that contain up to fifty speakers [1].

The current main spatialization technique consists in associating a fader of the mixing console to a single speaker. Then, the performer plays with the output level of each speaker. They actually encounter issues with complex spatialization transitions, as ten fingers cannot simultaneously control many faders.

The main idea is to introduce multi-point touch screens to replace hardware mixing consoles. The MIAM Spat software draws surfaces on a touch screen, and each surface represents a specific soundscape. A spatialization performance then becomes an interaction between these surfaces and the player's fingers.

The software described in this paper shows encouraging results, and is still evolving depending on artists' wishes. New possibilities and representations are offered, and MIAM Spat can be easily integrated to big spatialization sound systems.

1. INTRODUCTION

While commercial music production is released in stereophonic format, some composers prefer to add new dimensions to their pieces using spatialization systems. The biggest installations can reach between forty and fifty speakers in one concert hall ([1], [4]).

The Multi-touch Interface for Acousmatic Music – Spatialization (MIAM Spat.) project allows music spatialization using a multi-point touch interface. The main goal is to provide new opportunities to spatialization performers, who are using basic hardware mixing consoles at the moment. Touch technologies can help spatialization systems becoming more dynamic and intuitive.

This paper begins with a state-of-the-art review of existing spatialization techniques. A typical installation is presented, and the current use of such installations is described [7]. Then, some new desired features and characteristics are detailed, as current systems limit live performances' possibilities. These features and characteristics had been expressed in collaboration with an association of performers and composers.

Relying on multi-touch screens integration, the main new ideas are described in section 3. The MIAM Spat software's realization and first usage results are later explained.

2. STATE-OF-THE-ART REVIEW

Spatialization is in this paper the art of rendering music with complex and particular loudspeakers systems. This section describes the sound systems themselves, and how performers generally use them to produce audio effects.

2.1 Sound spatialization requirements

2.1.1 Original music pieces

Spatialization methods can be applied to various types of music pieces: spatialization systems can handle anything from a stereophony, up to a 32-track input.

According to a recent study conducted by Peters, Marentakis and McAdams [1], spatial aspects in music are mostly used "to enhance the listening experience", or "as a paradigm for artistic expression". Depending on composer's choices and goals, different sound entities could be mixed on a same track, or on several different tracks. All systems described afterwards are able to handle an arbitrary number of sound inputs.

2.1.2 Acousmonium

The MIAM Spat project aims at specific sound systems; it is of interest only if several distinct audio channels are available. This project focuses on interfacing with a sound system called *acousmonium*. Such systems are available at Musiques & Recherches [2] near Brussels, or at Groupe de Recherches Musicales [3] in Paris. The Birmingham ElectroAcoustic Sound Theatre (BEAST) is another example of such a sound system [4].

As defined firstly by F. Bayle in 1974 [5], an acousmonium is a "speaker orchestra" including at least sixteen speakers. These speakers are located in a three-dimensional space, and with various locations and orientations. Moreover, different kinds of speakers must be represented within an acousmonium: they must have different shapes, sizes, and spectral characteristics.

The objective is to be able to produce various sound effects, since acousmatic music focuses on the sound itself and not on musical instruments [3].

2.1.3 Typical installation

Based on several acousmoniums ([2], [3], [4]), the general organization of a sound system for spatialization is

represented on figure 1. Let N be the number of input tracks, and $M \geq N$ be the number of outputs to speakers.

Spatialization is basically the routing of N input tracks, to M speakers. This routing is actually done at the first spatialization step, within the playback software, because a dematerialized routing is easier to set up than a wired one. The mixing console contains M faders, and each fader is associated to a single speaker: this console then allows more precise live spatialization effects.

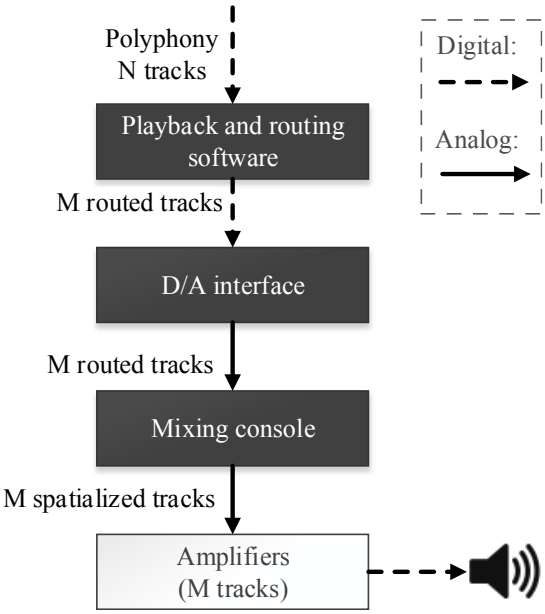

Figure 1. Current typical spatialization system.

For example, let n be one of the N input tracks. This track n may in practice be routed to all output tracks that are linked to a speaker on the ceiling. In this case, without a mixing console, this track n would play on the whole ceiling during the entire music piece. The mixing console allows to attenuate the signal from track n, before sending it to a particular speaker. For example, with a mixing console, the track n could be heard only at the front of ceiling, or only at its back.

2.2 Current techniques

2.2.1 Simulated spatialization for a few speakers

The most common spatialization technique is to modify instruments and sounds, and to route them into a tiny amount of channels. Psychoacoustic knowledge and physical simulation are employed in order to recreate spatialization effects [6]. The goal is to make these effects reproducible with accessible and conventional audio systems – such as stereophonic or 5.1 systems.

These kinds of processes are the opposite of spatialization with an acousmonium: a speaker orchestra renders effects into the real acoustic world. However, it should be noticed that most of the research and software production on the topic "spatialization" deals with this case.

2.2.2 Software spatialization

Given the diagram of figure 1, live spatialization could be considered from two different points of view. The spatialization tool may be the Digital Audio Workstation (DAW), as well as the mixing console.

The first method is to use built-in DAW routing and spatialization tools: according to a survey conducted by Peters et al. [1], this method is used by 75 percent of the respondents. The generated piece of music may then be ready for playback on an acousmonium with a static mixing console configuration. Containing up to $N = 16$ channels, such polyphonies do not need a significant live spatialization performance.

2.2.3 Hardware spatialization

The second approach is to play back a stereophonic or quadriphonic composition, and to perform most of the spatialization work at concert time. This is done with a hardware mixing console, since working with a DAW is not sufficiently intuitive and precise in a live situation – DAWs are basically controlled by a keyboard and a mouse. According to Peters et al. [1], 58 percent of spatialization artists use a hardware mixing console as a primary spatialization tool.

Composers who use hardware spatialization tend to export their pieces on stereo format, so that the spatialization is performed live [7]. The MIAM Spat interface will focus on this case – on console-based spatialization.

2.3 Current limitations

Playing on mixing consoles provides good precision about a single speaker's level, but complex sound transitions between speakers require a high level of skill.

Most of acousmoniums are built with stereophonic couples of speakers, and the left and right channels are placed side-by-side in the mixing console. To begin with a basic example, a tiny audio system of eight speakers is considered. They form a circle around the center of the concert hall, and the stereo couples are "Front", "Middle-front", "Middle-Back" and "Back". Playing a smooth sound translation from the back to the front of the room seems quite easy, with eight fingers. The player actually has to keep in mind four spatial entities: two close fingers modify the volume of a stereo couple.

Nonetheless, a smooth sound rotation around the center of the room requires more virtuosity. We will consider a rotation from the left to the right side, via the back side. To achieve this spatialization effect, many fingers have to move simultaneously and in opposite ways, as shown on figure 2.

Among several others, this example on figure 2 illustrates first limits of the mixing console, with displacement effects that are simple to conceive.

An acousmonium offers many other parameters, such as orienting sound directly towards audience or not. As described in section 2.1.2, speakers have different spectral characteristics, so that sound color is also a dynamic parameter. Timbre then becomes an important aspect of spatialization, and it can be controlled, as R. Normandeau explained [8].

Moreover, an acousmonium usually contains between $M = 20$ and $M = 50$ channels, which are far more than the hand's fingers. All these elements highlight a lack of versatile systems that help artists creating and realizing complex live spatialization operations.

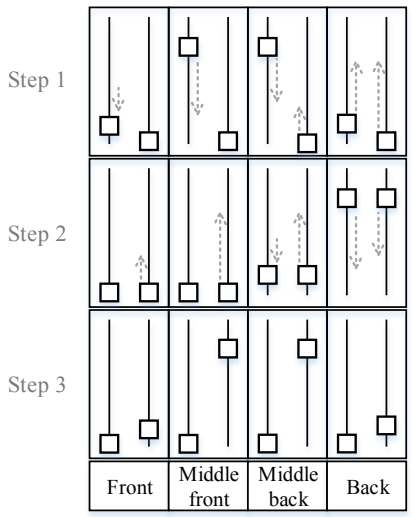

Figure 2. Smooth sound rotation: left - back - right. Spatialization performed on a system of eight speakers with a mixing console.

3. OBJECTIVES

3.1 Constraints

3.1.1 RME audio interfaces

The first specification is the use of RME [9] audio interfaces as digital to analog converters. The main reason is that developing prototypes on a precise hardware configuration is easier at the beginning. Besides, RME interfaces implement an intern matrix mixer associated to a software controller called Total Mix [9]. Matrix mixing principles are described in section 3.2.2.

3.1.2 Easy integration

It is also important that the new spatialization interface could be easily integrated to any current acousmonium. The setting up of such an audio system is already complicated, and a new performance instrument involving much more hardware would not be used in practice. Moreover, the goal is not to design a whole new spatialization system but rather to offer new possibilities using the existing system.

3.1.3 System latency

Latencies considered are the delays between a gesture on the spatialization controller, and its perceived consequences on output sound. The controller may be an analog console as well as a multi-touch screen. Issues have been encountered with previous MIAM Spat prototypes, since the project consists in a transition from a mechanical interface to a virtual computed one. MIAM Spat's reactivity is a crucial point, as previously expressed by performers in Brussels. An analog controller basically provides a zero-latency, while software controllers could lead to significant audio delays.

3.2 Main ideas

3.2.1 Multi-touch interfaces

With the growing amount of mass-produced multi-touch screens, the idea of using them emerged in collaboration with Musiques & Recherches. Such interfaces could replace mixing consoles during live spatialization performances. In theory, a ten-point touch interface is able to perform the same actions than a mixing table – with a virtual graphical mixing console for example – while new functionalities could be implemented.

There remain indeed differences between a mechanical mixing system and a virtual one. However, the consoles used in live performances by Musiques & Recherches do not contain motorized faders; the only feedback is the feeling of a cursor's position. Multi-point touch interfaces then should not be inconvenient.

3.2.2 Mixer configurations

As detailed in section 2.3, substantial improvements in intricate live spatialization transformations are yet possible. In this section 2.3, the left-back-right rotation is actually described with several steps; each one represents a static state of the whole hardware mixer. Performing spatialization is then driving the system from one state to another. These steps are software-managed in MIAM Spat, since most DAWs and RME's Total Mix include controllable mixing consoles.

Producing dynamics includes two phases. The first is to define mixer states, and interpolation methods between them. This phase occurs before the performance, which is the second phase consisting in controlling interpolations between chosen mixer states.

Transformations between mixer states are *controlled* by the performer, but they are *computed* by the MIAM Spat software. This idea solves the issue of transitions being too complex.

3.2.3 Matrix-based mixing

Let $\boldsymbol{I}, \boldsymbol{Q}$ be the input and output vectors of a spatialization mixer, both of size M. Spatialization using a mixing console actually consists in computing:

$$\boldsymbol{Q} = R(t)\boldsymbol{I} \tag{1}$$

Where $R(t)$ is the M-by-M routing matrix at time t. Please notice that a $R(t)$ matrix is the formal representation the mixer state at time t.

On the one hand, this routing matrix is diagonal when using an analog hardware mixing console: input m can only be routed to output m, with an attenuation coefficient. On the other hand, modern software mixers allow matrix routing, which formally means that $R(t)$ must not be a diagonal matrix. Any input can then be routed to any output and this brings new spatialization possibilities. A detailed in section 3.1.1, this routing feature is available with RME sound interfaces that we use.

3.2.4 Touch areas

Spatialization using MIAM Spat relies on the $R(t)$ matrix, but its dimension can reach $M = 40$ or $M = 50$: drawing such a matrix on a screen does not make sense, for a live performance. The choice has been made that important states – defined before the concert – are represented by surfaces on a touch screen. Those surfaces will be *convex*, and their borders will be *polygons*. A transition from a mixer state to another is then equivalent to a finger movement on the touch screen from a surface to another. An illustration of these touch areas is available on figure 3.

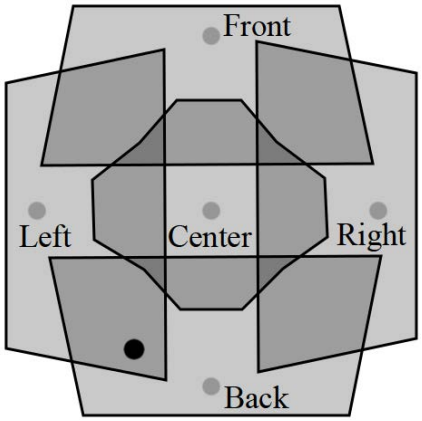

Figure 3. Basic example of MIAM Spat touch areas, representing mixer spatialization states. The black dot stands for the control finger's location.

On this figure 3, a finger touches the screen at the black dot's position at time t. This point belongs to two areas: one represents the "Back" spatialization state, and the other represents the "Left" state. MIAM Spat will then interpolate a routing matrix $R(t)$, given R_{back} and R_{left}. For more explanation on transition's computation, please see section 4.2.

The transition described on figure 2, section 2.3, is much easier to perform using MIAM Spat. The "Left-Back-Right" movement on a hardware mixer is equivalent to a simple finger displacement on the touch screen: a finger has to move from the "Left" labeled dot, to the "Back", and eventually to the "Right". During the whole gesture, MIAM Spat computes transitional $R(t)$ matrices.

4. REALISATION

4.1 Platform, and touch data

4.1.1 C# software

As latency is a critical point, the last MIAM Spat version is not developed on usual prototyping environments like Max [10] or Processing [11]. A lower-level and object-oriented programming language is necessary; C# has been chosen over C++ because of features making prototyping easier [12]. The last MIAM Spat software is then at the moment developed on Windows only.

4.1.2 Multi-touch data gathering

Whereas programming environments allow easy access to computer's mouse information, multi-point touch interaction data require a specific Application Programming Interface (API). Accessing such data is quick and robust using C# and Windows API [12].

The use of a recent protocol for multi-touch interfaces was also considered: some tests were made using TUIO ([13], 2005), which is open-source and platform-independent. Latency issues were unfortunately encountered, since many multi-touch screens do not send native TUIO messages.

Please notice that one touch point is sufficient to control the current MIAM Spat interface, but features using several touch points are planned: this is why multi-point touch support is already necessary.

4.2 Interpolation methods

4.2.1 Interpolation functions

Details on interpolation methods will rely on the spatialization example from figure 3. At first, a basic sound displacement from the left to the back of the concert hall will be explained.

At the beginning, the finger is placed near the grey dot labeled "Left"; this dot locates an arbitrary mass center for the "Left" area. Let w_{left} be a coefficient computed to represent how the finger is close to the mass center. The exact method for w_i computing – for a given area i – is given in section 4.2.2, but the main constraints are:

$$\begin{cases} w_i = 1.0 \text{ when a finger is on mass center} \\ w_i = 0.0 \text{ when a finger is on area's border} \end{cases} \quad (2)$$

When the finger is only over the "Left" area, the routing matrix is $R(t) = R_{left}$. Then, the finger moves over both areas "Left" and "Back", and MIAM Spat computes two coefficients w_{left} and w_{back}. Interpolated routing matrix depends on these two coefficients, and also on the two routing matrices R_{left} and R_{right}, such that:

$$R(t) = f(R_{left}, w_{left}, R_{back}, w_{back}) \quad (3)$$

The function f might be an arbitrary interpolation, and we choose for MIAM Spat a linear interpolation. When a finger moves over at least one touch area, the general formula is:

$$R(t) = \frac{\sum_i w_i R_i}{\sum_i w_i} \quad \text{where } i \text{ is a touch area} \quad (4)$$

4.2.2 w_i coefficients computing

A coefficient w_i expresses how close a finger is to the mass center of touch area i: w_i will be called *interaction weight*. To satisfy constraints (2), various functions may be employed, but we chose to use a projection of the finger touch point.

Let G_i be the center of mass of touch area i, and $A_{i1}, \ldots, A_{iK}$ the K points that form the polygonal border of

surface i. Let T be the finger interaction point. Touch areas in MIAM Spat are convex surfaces, and their borders are polygons: this is why a touch area can be divided into K triangles $G_iA_{i1}A_{i2}, \dots, G_iA_{iK}A_{i1}$.

If a touch point T is over the area i, it necessary belongs to exactly 1 of the K triangles. Let T' be the intersection point between G_iT, and the side of that triangle that is a border of the touch area. Figure 4 illustrates this geometry construction, with a surface similar to the "Left" surface from figure 3.

Given the point T', the interaction weight w_i is:

$$w_i = \frac{\|\overrightarrow{G_iT}\|}{\|\overrightarrow{G_iT'}\|} \quad (5)$$

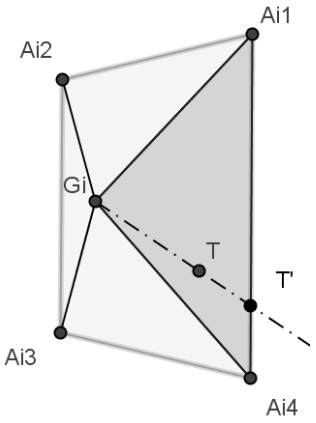

Figure 4. Interaction weight computing by projection, with K=4. G_i is the center of mass, T is the touch point, T' is the projected point.

This projection system allows continuous sound transitions when moving from an area to another. When leaving an area i, the computed interaction weight w_i tends to zero near the border, so that contribution from area i to the routing matrix $R(t)$ becomes negligible.

4.3 Communication means

4.3.1 Interfacing

After updating the routing matrix $R(t)$, the result is sent to RME Total Mix, which is a mixer software. It controls intern matrix mixer of RME sound interfaces, and necessarily runs on the computer which plays back an acousmatic composition.

MIAM Spat software runs on a separate Windows computer, connected to the computer running Total Mix. Figure 5 shows where the MIAM Spat interface is connected, and should be compared to figure 1.

4.3.2 Total Mix MIDI controlling

Matrix mixer data was initially sent using the Musical Instrument Digital Interface (MIDI) protocol, but a bandwidth issue was encountered. A MIDI connection offers a bandwidth of only 3,125 bytes per second [14], while sending M-by-M matrices requires much more.

A common size for an acousmonium is $M \geq 40$, then a R(t) matrix may be more than 1600 bytes. The amount of data actually sent may be smaller, as each $R(t)$ element is not necessarily modified by a transition. However, the latency could reach more than 500ms in the worst case, which does not satisfy our initial constraints.

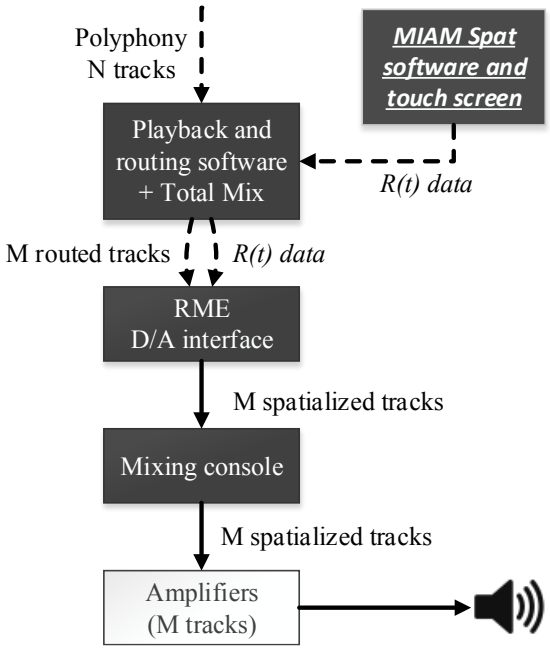

Figure 5. New spatialization system, including the MIAM Spat interface.

When using the MIAM Spat interface, the mixing console becomes useless and has to be in "neutral" position: all its faders will have to be at the $0dB$ position.

4.3.3 OSC via local network

Open Sound Control (OSC) protocol allows data transmitting over UDP packages [15], which is much faster and offers a bigger bandwidth. So, the computer running MIAM Spat is connected by Ethernet to the computer running Total Mix.

5. FIRST RESULTS

Two main categories of results are obtained with MIAM Spat: at first the satisfaction of initial given constraints, and later new features available for live performances.

5.1 Constraints

5.1.1 Latency

Results about latency are hard to quantify precisely: this would require a system that computes the exact delay between a tap on the touch screen and its audio consequences.

The overall latency is nonetheless weak. When using MIAM Spat with a mouse – and not with a finger on a touch screen – delays are not significant. The audio system *seems* to follow the exact behavior of the mouse.

Latency is however felt when using multi-touch screens. Before sending high-level touch information, these screens need a slight delay to process raw touch data from sensors. Depending on the screen's technology and manufacturer, this processing may be negligible or not. Some multi-point touch screens showed very good reactivity, while others lead to perceptible delays.

5.1.2 Integration to current systems

As expressed in section 3.1, the ease of integration to an acousmonium is important. The interfacing scheme from section 4.3.1 shows that MIAM Spat is connected to only one of the spatialization chain's elements.

This makes its integration better, as it can be easily removed if not used by a performer. Besides, MIAM Spat sends data, but does not need to receive any kind of data. The current spatialization system then remains exactly the same, with or without MIAM Spat.

5.2 New spatialization abilities

5.2.1 Visualization

The most obvious new feature with MIAM Spat, is that spatialization can be graphically represented on a surface. A touch area's location may be related to its associated mixer configuration, and its shape can be freely defined – it only has to be convex.

For example, a touch area for the "Front Direct" mixer configuration could be located at the top of the touch screen, near its center. A touch area for a "Right Diffuse" mixer routing configuration could be located at the right of the screen, near its frame.

Traditional spatialization using hardware mixing consoles did not allow such representations. One routed track, among the M tracks, could only get a label on the mixing console.

5.2.2 Complex, smooth, and fast transitions

As quickly detailed in section 3.2.2, MIAM Spat technically allows complex spatialization transitions. Applying sound displacements and effects becomes easier and more intuitive. A precise spatialization control requires however to setup many different touch areas, in order to get many different available mixer configurations.

The computing method for interaction weights w_i also allows intricate transitions to be smooth and continuous. As detailed in section 4.2.2, the projection method ensures sound continuity when going from a touch area to another – from a mixer state to another. Moreover, MIAM Spat allows performers to play the transition dynamics that they desire. The linear interpolation method is the most simple and the most neutral, so that performers are not influenced when playing.

Very fast transitions can nonetheless occur, and that was not possible with a hardware mixing console. The first way this is achieved, is to quickly move a finger from one touch area to another. The second way is to tap the touch screen at any location. When the tap hits one or several touch areas, a new routing state is instantaneously computed and sent to Total Mix. This results in an instantaneous audio spatialization change, whereas this was not possible with a mixing console.

6. CONCLUSIONS

Early MIAM Spat usage results are encouraging, as they fulfill initial objectives and constraints. At first, MIAM Spat is actually designed to easily integrate an acousmonium, since a performer can choose to use it or not. Introducing a new artistic tool requires times, and some people may prefer to use the traditional hardware mixing console.

The use of multi-point touch interfaces brings new spatialization possibilities, as it was one of the main goals of the MIAM Spat project. It introduces a new approach of spatialization, based on transitions between mixer states. This approach must still be tested and commented on by more performers, in order to improve it, but it is already usable.

Many features are however still being defined and developed in association with Musiques & Recherches. Among them, the main one is to create automatic transitions, based on physical models. The movements could for example include inertia parameters, since current spatialization dynamics in MIAM Spat are exclusively controlled by the performer.

Acknowledgements

We would like to offer our special thanks to Ludovic Laffineur (ISIB – LARAS) for his useful comments on an earlier version on this manuscript. We also thank Annette Vande Gorne and Rafael Muñoz Gomez from Musiques & Recherches for their collaboration to the project.

7. REFERENCES

[1] N. Peters, G. Marentakis, and S. McAdams, "Current Technologies and Compositional Practices for Spatialization: A Qualitative and Quantitative Analysis," *Computer Music Journal*, vol. 35, no. 1, pp. 10-27, 2011.

[2] Musiques & Recherches association. "Dispositifs techniques," retrieved March 27, 2014. http://www.musiques-recherches.be/fr/concerts/dispositions-techniques

[3] M. Battier. "What the GRM brought to music: from musique concrete to acousmatic music," *Organised Sound*, vol. 12, no 3, p. 189, 2007.

[4] University of Birmingham, "Studios, Facilities and Equipment," retrieved on March 30, 2014. http://www.birmingham.ac.uk/facilities/BEAST/studios/index.aspx

[5] F. Bayle, "A propos de l'acousmonium," *Recherche Musicale au GRM*, vol. 397, pp. 144-146, 1989.

[6] M. Schumacher and J. Bresson, "Spatial Sound Synthesis in Computer-Aided Composition," *Organised Sound*, vol. 15, no 3, pp. 271-289, 2010.

[7] H. Tutschku, “On the interpretation of multi-channel electroacoustic works on loudspeaker-orchestras: some thoughts on the GRM-acousmonium and BEAST,” *Journal of Electroacoustic Music*, vol. 14, 2002.

[8] R. Normandeau, “Timbre Spatialisation: The medium is the space,” *Organised Sound*, vol. 14, pp 277-285, 2009.

[9] RME manufacturer website. http://www.rme-audio.de/en_index.php

[10] Max software website. http://cycling74.com/products/max/

[11] Processing software website. http://www.processing.org/

[12] Microsoft Developer Network Website. “Why Use C#,” retrieved on March 30, 2014. http://msdn.microsoft.com/en-us/library/aa664274%28v=vs.71%29.aspx

[13] M. Kaltenbrunner, T. Bovermann, R. Bencina, E. Costanza, "TUIO - A Protocol for Table-Top Tangible User Interfaces," in *Proc. of the 6th Int. Workshop on Gesture in Human-Computer Interaction and Simulation (GW 2005)*, Vannes, France, 2005. http://www.tuio.org

[14] MIDI Manufacturers Association, “Tech Specs & Info,” retrieved on March 30, 2014. http://www.midi.org/techspecs/

[15] M. Wright, “Open Sound Control-A New Protocol for Communicating with Sound Synthesizers,” in *Proc. of the 1997 Int. Computer Music Conference,* 1997, pp. 101-104.