MashtaCycle: On-Stage Improvised Audio Collage by Content-Based Similarity and Gesture Recognition

Christian Frisson[2,⋆], Gauthier Keyaerts[1], Fabien Grisard[2,3], Stéphane Dupont[2], Thierry Ravet[2], François Zajéga[2], Laura Colmenares Guerra[2], Todor Todoroff[2], and Thierry Dutoit[2]

[1] aka Very Mash'ta and the Aktivist, artist residing in Brussels, Belgium
http://www.mashtacycle.be
[2] University of Mons (UMONS), numediart Institute
Boulevard Dolez 31 B-7000 Mons Belgium
http://www.numediart.org
[3] ACROE / Institut Phelma, Grenoble, France
christian.frisson@umons.ac.be

Abstract. In this paper we present the outline of a performance in-progress. It brings together the skilled musical practices from Belgian audio collagist Gauthier Keyaerts aka *Very Mash'ta*; and the realtime, content-based audio browsing capabilities of the *AudioCycle* and *LoopJam* applications developed by the remaining authors. The tool derived from *AudioCycle* named *MashtaCycle* aids the preparation of collections of stem audio loops before performances by extracting content-based features (for instance timbre) used for the positioning of these sounds on a 2D visual map. The tool becomes an embodied on-stage instrument, based on a user interface which uses a depth-sensing camera, and augmented with the public projection of the 2D map. The camera tracks the position of the artist within the sensing area to trigger sounds similarly to the *LoopJam* installation. It also senses gestures from the performer interpreted with the *Full Body Interaction (FUBI)* framework, allowing to apply sound effects based on bodily movements. *MashtaCycle* blurs the boundary between performance and preparation, navigation and improvisation, installations and concerts.

Keywords: Human-music interaction, audio collage, content-based similarity, gesture recognition, depth cameras, digital audio effects.

1 Introduction

Since the advent of affordable signal sensing and processing, ubiquitous and social networks, massive crowd-sourced multimedia datasets are being enriched everyday. These technologies allow audio artists to easily create their sounds or source these elsewhere, digitally, from the "ocean of sounds" [19].

⋆ Corresponding author.

M. Mancas et al. (Eds.): INTETAIN 2013, LNICST 124, pp. 114–123, 2013.

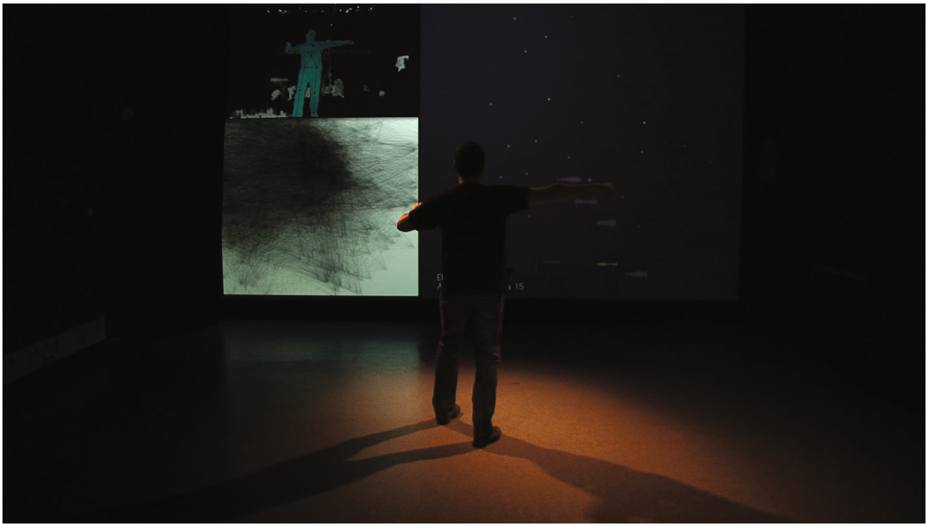


Fig. 1. Picture of an early version of stage setup at the numediart Institute where Gauthier Keyaerts interacts with *MashtaCycle* by his position tracked by a Kinect sensor. In the top left corner of the screen his segmented body as sensed by the Kinect through *OpenNI/NiTE* appears in the *FUBI* view. In the down left corner, visuals designed by François Zajéga are played back, acting as visual score. In the right column, a collection of sounds is visualized and rendered with the *MediaCycle* framework. Picture courtesy of Laura Colmenares Guerra (http://www.ulara.org).

Western digital music often relies on scores to transcribe and describe musical pieces. Here we consider sound samples as vocabulary, as do musical genres such as hip-hop, DJ'ing, electro-acoustic music. The map of sounds visualized in the *MashtaCycle* instrument becomes the score.

However, the emphasized musical expression in terms of sound generation offered by computer music suffers from an important drawback: the control of the sound generation. The NIME and ICMC conference communities, for instance, have been focusing on addressing this issue since decades [1]. William Brent proposes a fully opensource pipeline for content-based audio browsing through free-form gestural control [4] that seems suitable for prototyping, while Vigliensoni digs deeper into the technologies offered for gestural control and sound synthesis [12]. We aim at offering a musical instrument that can complement the advantages offered by tangible and free-form interfaces [7].

In section 2, we describe the architecture of *MashtaCycle*. In section 3, we provide an overview of our accomplishments through this project and open with perspectives towards improvements, not without acknowledging (in section 3) all the people that made this project feasible.

2 Architecture

Figure 2 illustrates the architecture of the *MashtaCycle* system: sound files are first imported to be organized by content-based similarity as described in subsection 2.1, so as to be navigable in a 2D-space. To make the audio browsing more performative, simple gesture recognition techniques are made use of as explained in subsection 2.2. Mappings are drawn from gestures not only to sound rendering cues and effects (subsection 2.3), but also to generative visuals (subsection 2.4). The prototyping method that helped to refine the mappings up to an hardcoded system is discussed in subsection 2.5. Subsection 2.6 closes the architecture description by an opening: *MashtaCycle* can be hybridized from different settings, as an installation and as a performative tool.

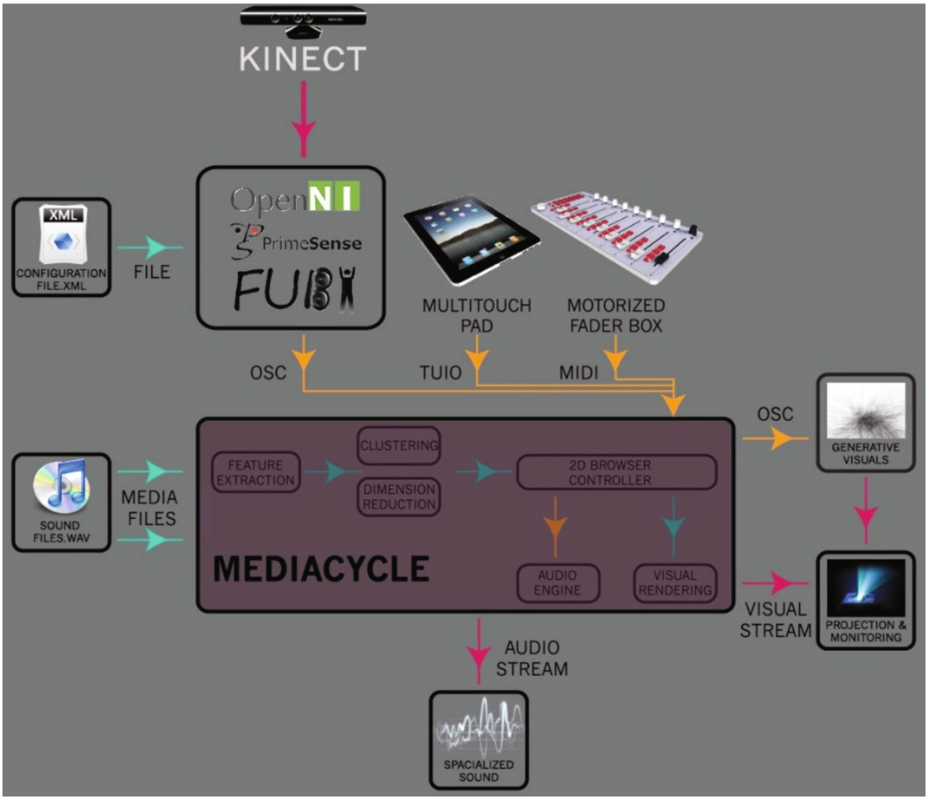


Fig. 2. Architecture of the current version of *MashtaCycle*

2.1 Content-Based Visual Audio Browsing

MashtaCycle is the next iteration of the *AudioCycle* series of applications, with the difference that it is tailor-made for an artist. *AudioCycle* is an application for organizing sounds by content-based similarity developed since 2008 in the numediart Institute of the University of Mons, described in [5], [9] then [8].

The notable difference since the last entertainment uses of *AudioCycle* is the choice of the algorithm that reduces the dimension of the audio feature space down to the 2D dimension of the visual space of the audio browser.

We would before use a simple algorithm named the "*Propeller*" that would, after a K-Means clustering step with a user-defined number k of clusters, create a visualization featuring a central symmetry, with k propels evenly-distributed angularly by the position of the centroid of their cluster, each node of each cluster positioned accordingly to its Euclidean distance to the centroid in the feature space. At the time it allowed to distribute sounds from the same instrument in a collection constituted of monophonic instrument loops in easily distinguishable areas, similarly to musicians in a rock band or a symphonic orchestra, as elaborated in [8]. This visualization was affected by several flaws:

- sounds weren't always properly clustered by instrument (or timbre), what is highly dependent on the number of clusters desired by the user versus the number of actual classes in the sound database;
- sounds were often occluding each others;
- an artifact inherent to its algorithm often made the visualization look more like the head of a string trimmer (gardening tool) than a propeller, sounds from the same cluster escaping the centroid in a curved line instead of being grouped in a propel.

After an evaluation of content-based visualizations by some of the authors [6], we changed our default visualization algorithm in favor of the Student-t distributed Stochastic Neighbor Embedding (t-SNE) algorithm and refined the choice of audio features used for the clustering and visualization, in short Mel-Frequency Cepstral Coefficients (MFCC) and their first- and second-order derivatives plus Spectral Flatness, using *Yaafe* [13]. For further explanation we direct the readers to the paper describing the evaluation [6]. It dramatically improves all the aforementioned issues (up to the accuracy unsupervised content-based organization can offer), notably by providing a layout where visual neighbors are separated by a repeatable distance, reducing overlap and easing the navigation though less closest node jitter.

In [11], the authors of the *CataRT* Max/MSP based environment for concatenative synthesis discuss other strategies for visual mapping.

2.2 Gesture Recognition

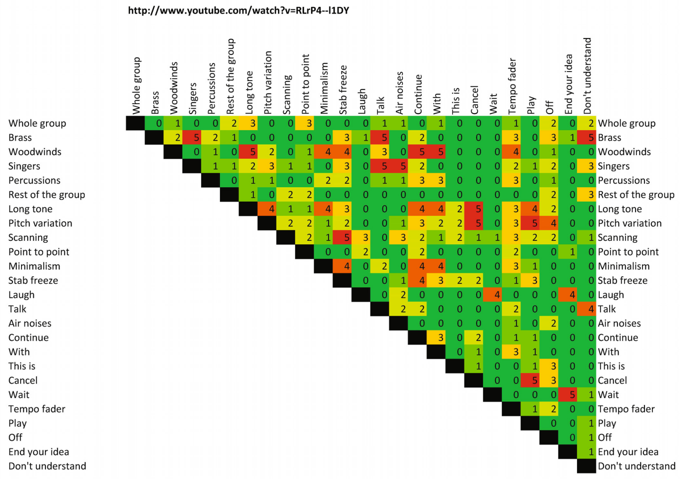
LoopJam [8], an installation made with *AudioCycle* controlled by a Kinect camera, allowed visitors to trigger sounds by their position in front of the projected map of sounds. At the time, it used Daniel Roggen's *QtKinectWrapper* (https://code.google.com/p/qtkinectwrapper/), that we modified to send the 2D position in the plane of the floor of all users through OpenSoundControl (OSC). We aim at turning this installation into a musical instrument for a single user. To do so, we needed to offer more control on the sound rendering than just looped playback activation of the closest node, still through gestures that would be sensed with a Kinect camera.

To our knowledge, not many "plug and play" solutions are available for gesture recognition using the Kinect. The *Kinect SDK* from Microsoft may provide gesture recognition methods, but since it is available only on Microsoft platforms, it was instantly discarded for our use. To name a few, *Kinectar* (http://ethnotekh.com/software/kinectar/) provides heuristics-based gesture recognition (variation of distances between joints), it has the advantage of being designed by an artist, Chris Vik, for himself, what goes beyond the growing Kinect hacks, but it relies on many dependencies including closed-source ones. *XKin* was nominated for the Open Source Software Competition of ACM Multimedia 2012 [14]. It uses Hidden Markov Models (HMM) for hand poses recognition. While it offers a fully opensource pipeline with *libfreenect* instead of PrimeSense's *OpenNI/NiTE* combination that prevents a "drag-and-drop" installation and distribution since *NiTE* is closed-source, it is for now restricted to hand gestures, rather than full body gestures.

We chose to fork the *Full Body Interaction Framework (FUBI)* [10] (http://www.hcm-lab.de/fubi.html), opensource, from the University of Augsburg, which supports four gesture categories: 1) static postures, 2) gestures with linear movement, 3) combination of postures and linear movement and 4) complex gestures. In addition, the framework enables to detect the number of fingers the users are showing in front of the sensor, but it requires users to keep a fixed distance to the sensor. We added an *OpenSoundControl (OSC)* bridge to *FUBI* so as to directly communicate with our *MediaCycle* application. For that purpose we used the lightweight *OscPkt* library (http://gruntthepeon.free.fr/oscpkt/).

In the last chapter of his book [15], Dan Saffer catalogs free-form gestures and movements for the design of gestural interfaces and provides insight on references diving further into the topic, in the field of Human-Computer Interaction. Much research have been performed on musical gestures [21], borderline between art, science and technology.

Sound painting, originated by Walter Thompson in 1974 [17] and later popularized by John Zorn (notably his *Cobra* series), allows a conductor to have an orchestra of musicians improvise music by communicating through a sign language featuring hundreds to thousands of signs, in categories such as "who" (instrument or musicians), "what" (musical content and processing), "how" and "when". Figure 3 illustrates how sound painting inspired us through the design process.



http://www.youtube.com/watch?v=RLrP4--l1DY

	Whole group	Brass	Woodwinds	Singers	Percussions	Rest of the group	Long tone	Pitch variation	Scanning	Point to point	Minimalism	Stab freeze	Laugh	Talk	Air noises	Continue	With	This is	Cancel	Wait	Tempo fader	Play	Off	End your idea	Don't understand	
Whole group		0	1	0	0	2	3	0	0	3	0	0	0	1	1	0	1	0	0	0	1	0	2	0	2	Whole group
Brass			2	5	2	1	0	0	0	0	0	3	1	5	0	2	0	0	0	0	3	0	3	1	5	Brass
Woodwinds				0	1	0	5	2	0	1	4	4	0	3	0	5	5	0	0	0	4	0	1	0	0	Woodwinds
Singers					1	1	2	3	1	1	0	3	0	5	5	2	0	0	0	0	2	1	2	0	3	Singers
Percussions						0	1	1	0	0	2	2	0	1	1	3	3	0	0	0	3	0	1	0	0	Percussions
Rest of the group							1	0	2	2	0	0	0	0	0	0	0	0	0	0	0	0	2	0	3	Rest of the group
Long tone								4	1	1	4	3	0	0	0	4	4	2	5	0	3	4	2	0	0	Long tone
Pitch variation									2	2	1	2	0	0	1	3	2	2	5	0	3	5	4	0	0	Pitch variation
Scanning										2	1	5	3	0	3	2	1	2	1	1	3	2	2	0	1	Scanning
Point to point											0	0	2	0	0	2	0	0	0	0	2	0	0	1	0	Point to point
Minimalism												4	0	2	0	4	4	0	0	0	3	1	0	0	0	Minimalism
Stab freeze													0	0	1	4	3	2	2	0	1	3	0	0	0	Stab freeze
Laugh														0	2	0	0	0	0	4	0	0	0	4	0	Laugh
Talk															2	2	0	0	0	0	2	0	0	0	4	Talk
Air noises																0	0	0	0	0	1	0	2	0	0	Air noises
Continue																	3	0	2	0	1	1	0	0	0	Continue
With																		0	1	0	3	1	0	0	0	With
This is																			1	0	0	1	3	0	0	This is
Cancel																				0	0	5	3	0	0	Cancel
Wait																					0	0	0	5	1	Wait
Tempo fader																						1	2	0	0	Tempo fader
Play																							0	0	1	Play
Off																								0	1	Off
End your idea																									1	End your idea
Don't understand																										Don't understand

Fig. 3. Subjective evaluation by one of the authors of a subset of the Sound Painting gestures explained in Thomas Claus' video (http://www.youtube.com/watch?v=RLrP4--l1DY). A first selection step keeps the gestures that seemed to be detectable using a Kinect sensor. A second step rates these on a [0; 5] integer Likert-like scale that estimates the probability of crosstalk between each possible couple, presented as a confusion matrix, with cue times of gestures in the video.

2.3 Sound Playback, Synthesis and Effects

A lot of research has been performed on the control of digital audio effects, either by direct gestural control or by adaptive control based on the content-based sound features [20]. Rather than adding support for usual audio effects plugin software development toolkits such as *VST* or *AudioUnits* to iterate over the *LoopJam* [8] installation, we created a new audio engine based on the *Synthesis Toolkit (STK)* [16] (`https://ccrma.stanford.edu/software/stk/`) itself built upon the *RtAudio* backend and providing classes for common audio effects (chorus, delay, reverb, etc...), since this solution is suitable for fast prototyping and avoids potential issues generated by third party plugins.

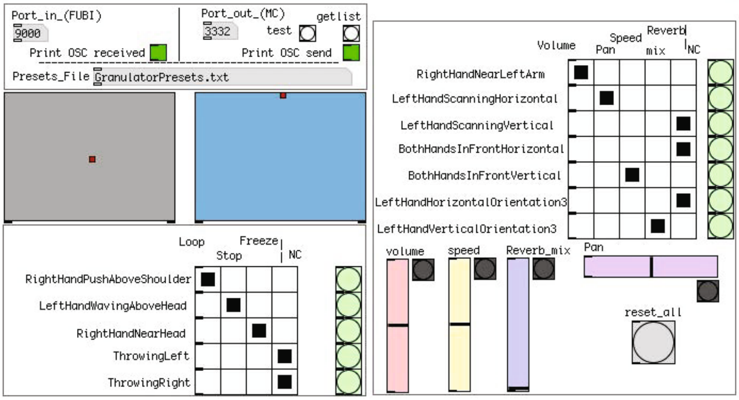
2.4 Visual Rendering

The monitoring views offered by the *MediaCycle* browser and the *FUBI* gesture recognition tool are mandatory for using the system as a musical instrument, but these don't help the system to qualify properly as a piece of interactive arts which would offer a more poetic visualization. A sound-dependent visual rendering is being prototyped by Belgian artist François Zajéga (`http://www.frankiezafe.org`), In Figure 1 extracted from the demo video of the *MashtaCycle* project, visuals created by François Zajéga were played back in a loop as a movie file so as to influence the audio collage improvisation by Gauthier Keyaerts, similarly to a graphical score.

A visual rendering that reacts to the sound content improvised by Gauthier Keyaerts is in progress. To do so, audio features will be extracted from the sound rendering after the effects processing. While the *MediaCycle* framework doesn't yet support realtime stream analysis, an intermediate prototyping solution is offered by William Brent's *timbreID* objects for *PureData* [4].

2.5 Mapping, Prototyping

Building upon earlier works with *MediaCycle* [9], we decided to apply the same fast user interface prototyping method using the *PureData* environment. Lots of mapping-related steps affect the architecture of *MashtaCycle*: 1) audio features to 2D space, 2), gestures to audio effects, 3) post-processed audio rendering to visuals. For now one-to-one mappings are being used, except for step 1). Figure 4 illustrates an example of mapping.



Port_in_(FUBI) 9000

Print OSC received 🟩

Port_out_(MC) 3332

test ○ getlist ○

Print OSC send 🟩

Presets_File GranulatorPresets.txt

	Loop	Stop	Freeze	NC
RightHandPushAboveShoulder	■			
LeftHandWavingAboveHead		■		
RightHandNearHead			■	
ThrowingLeft				■
ThrowingRight				■

	Volume	Pan	Speed	Reverb mix	NC
RightHandNearLeftArm	■				
LeftHandScanningHorizontal		■			
LeftHandScanningVertical					■
BothHandsInFrontHorizontal					■
BothHandsInFrontVertical			■		
LeftHandHorizontalOrientation3					■
LeftHandVerticalOrientation3				■	

Fig. 4. *PureData* patch for the 1-to-1 mapping of gestures to sound effects

After the prototyping phase, mappings have been hard-coded in C++ temporarily inside the *FUBI* fork so as to reduce the number of applications to launch and monitor (what we do with *Lingon* (`http://sf.net/p/lingon/`), a Mac OSX GUI for *launchd*), and the overhead of the *PureData* application.

2.6 Hardware and Setup Requirements

The architecture of this project has been sketched over the last subsections.

Since the depth sensing camera emits a grid of infrared beams and receives its reflections from the scene, stage lights have to be dimmed properly or equipped with filters (infrared, or blue colored) so as not to interfere with the sensing.

Many loudspeaker configurations may be used. For a good sound immersion we would use at least a cost-effective quadraphony (two loudspeakers surrounding the stage, another two behind the audience).

Multiple screen configurations are possible, along what the artist and the audience may see, with the following views activated or not: 1) *FUBI* gesture recognition, 2) *MediaCycle* visual audio browsing, 3) (audio-dependent) visuals. Different setups may be presented so as to suit venues, for instance:

1. demo: the actual setup illustrated in Figure 1
2. concert: the audience only sees the audio-dependent visual rendering on a large screen projection, the artist monitors the *FUBI* gesture recognition and the *MediaCycle* audio browsing views from a second projected screen located in one side of the stage

An innovative feature of this work is that after a concert, the musical instrument can be turned into an installation that the audience can visit to better understand it while chilling out and discussing with the crew of this project.

3 Conclusion, Perspectives

We designed a first prototype of a musical instrument tailored for an artist wanting to create improvised audio collages, using gesture recognition through a depth sensing camera and content-based visual audio browsing.

Our fork of the *Full Body Interaction Framework (FUBI)* with *OpenSoundControl (OSC)* output and new gestures is available on `http://github.com/ChristianFrisson/FUBIOSC`, we hope most of it will be integrated back to the main distribution of *FUBI*, and we plan to integrate it as a *flext* object for *PureData* and *Max/MSP* into the *DeviceCycle* [9] distribution for fast user interface prototyping.

While this work satisfies the artist it is designed for, it still needs a quantitative evaluation. We tweaked the current gesture recognition technique up to the point that the artist using the tool feels that gestures are properly recognized most of the times. We may measure the repeatability in detecting a certain number of same instances of a given gesture, but we plan first to change the gesture recognition method in favor of another one from the field of machine learning such as Hidden Markov Models (HMM) [3,18], Dynamic Time Warping (DTW) [2]. The current method is not morphologically-independent and requires fiddling with numbers on an XML file, what we believe is not suitable for most potential users of this project, first and foremost the artist for whom it is designed, when new gestures are requested. Gesture design through recording is in progress.

Some of the authors, which are musicians of various levels of training, would prefer contact-based and/or tangible interaction over the free-form interaction offered in this project [9]. Gestural interfaces with haptic feedback are being prototyped, similarly to [22].

Acknowledgements. The majority of the authors are funded by a grant from the Walloon Region of Belgium (N°716631) that allowed to found the numediart Institute.

Gauthier Keyaerts obtained a grant from the *Digital Arts section* of *Fédération Wallonie-Bruxelles* (`http://www.arts-numeriques.culture.be`) that complements the financial coverage of the design phase of this project. We would like to thank the committee for having nominated our project and Anne Huybrechts for having advised us carefully while submitting our proposal.

Fabien Grisard obtained an ExploRA'sup grant from the French Region of Rhône-Alpes that covers parts of his visiting at the numediart Institute for his MSc graduation internship, focusing on the gesture recognition/design.

We are redeemable to Felix Kistler, from the University of Augsburg, Germany, for creating *FUBI*, releasing it open source (`http://sf.net/p/fubi`), and answering our related questions spot-on and almost in realtime.

We thank our colleagues Nicolas d'Alessandro and Joëlle Tilmanne for their expert insight on more solid gesture recognition methods, we hope to work with them towards an improved version of this project.

We thank the artistic associations and individuals that have been curating iterations of prototypes this project builds itself upon: Philippe Franck and the team of *Transcultures* (`http://www.transcultures.be`) plus the team of movie theater *Galeries* (`http://www.galeries.be`) hosting *LoopJam* for the *Citysonic* 2012 festival in Brussels (`http://www.citysonic.be`); the organizers of the Kikk festival (`http://www.kikk.be`), who welcomed us for the 2012 edition, Stephan Dunkelman, Marianne Binard and the team of *Halolalune* (`http://www.halolalune.be`) plus Wim Verhulst and the team of the *Musical Instruments Museum (MIM)* of Brussels (`http://www.mim.be`) for hosting *LoopJam* through the annual week of *La Semaine du Son* in late January 2013 (`http://www.lasemaineduson.be`).

References

1. Bekkedal, E.: Music kinection: Musical sound and motion in interactive systems. Master's thesis, Department of Musicology, University of Oslo (2012)
2. Bettens, F., Todoroff, T.: Real-time DTW-based gesture recognition external object for max/msp and puredata. In: Proceedings of the 6th Sound and Music Computing Conference, Porto, Portugal, July 23-25 (2009)
3. Bevilacqua, F., Zamborlin, B., Sypniewski, A., Schnell, N., Guédy, F., Rasamimanana, N.: Continuous realtime gesture following and recognition. In: Kopp, S., Wachsmuth, I. (eds.) GW 2009. LNCS, vol. 5934, pp. 73–84. Springer, Heidelberg (2010)
4. Brent, W.: Physical navigation of virtual timbre spaces with timbreID and DILib. In: Proceedings of the 18th International Conference on Auditory Display, Atlanta, GA, USA, June 18-21 (2012)
5. Dupont, S., Frisson, C., Siebert, X., Tardieu, D.: Browsing sound and music libraries by similarity. In: 128th Audio Engineering Society (AES) Convention, London, UK, May 22-25 (2010)

6. Dupont, S., Ravet, T., Picard-Limpens, C., Frisson, C.: Nonlinear dimensionality reduction approaches applied to music and textural sounds. In: IEEE International Conference on Multimedia and Expo (ICME), San Jose, USA, July 15-19 (2013)
7. Frisson, C.: Designing tangible/free-form applications for navigation in audio/visual collections (by content-based similarity). In: Graduate Student Consortium of the ACM Tangible, Embedded and Embodied Interaction Conference (TEI 2013), Barcelona, Spain, February 10-13 (2013)
8. Frisson, C., Dupont, S., Leroy, J., Moinet, A., Ravet, T., Siebert, X., Dutoit, T.: LoopJam: Turning the dance floor into a collaborative instrumental map. In: Proceedings of the New Interfaces for Musical Expression (NIME), Ann Arbor, Michigan, USA, May 21-23 (2012)
9. Frisson, C., Dupont, S., Siebert, X., Tardieu, D., Dutoit, T., Macq, B.: DeviceCycle: Rapid and reusable prototyping of gestural interfaces, applied to audio browsing by similarity. In: Proceedings of the New Interfaces for Musical Expression++ (NIME++), Sydney, Australia, June 15-18 (2010)
10. Kistler, F., Sollfrank, D., Bee, N., André, E.: Full body gestures enhancing a game book for interactive story telling. In: International Conference on Interactive Digital Storytelling, Proceedings of ICIDS 2011 (2011)
11. Lallemand, I., Schwartz, D.: Interaction-optimized sound database representation. In: Proceedings of the 14th International Conference on Digital Audio Effects (DAFx 2011), Paris, France, September 19-23 (2011)
12. Martin, A.G.V.: Touchless gestural control of concatenative sound synthesis. Master's thesis, McGill University, Montreal, Canada (2011)
13. Mathieu, B., Essid, S., Fillon, T., Prado, J., Richard, G.: Yaafe, an easy to use and efficient audio feature extraction software. In: Proceedings of the 11th ISMIR Conference, Utrecht, Netherlands (2010)
14. Pedersoli, F., Adami, N., Benini, S., Leonardi, R.: XKin: eXtendable hand pose and gesture recognition library for kinect. In: Proceedings of the 20th ACM International Conference on Multimedia (MM 2012), pp. 1465–1468. ACM, New York (2012)
15. Saffer, D.: Designing Gestural Interfaces. O'Reilly Media, Inc. (2009)
16. Scavone, G.P., Cook, P.R.: RtMidi, RtAudio, and a Synthesis Toolkit (STK) update. In: of the International Computer Music Conference (2005)
17. Thompson, W.: Soundpainting: The Art of Live Composition. In: Walter Thompson, 2006. with instructional DVD (2006)
18. Tilmanne, J.: Data-driven Stylistic Humanlike Walk Synthesis. PhD thesis, University of Mons (2013)
19. Toop, D.: Ocean Of Sound: Aether Talk, Ambient Sounds and Imaginary Worlds. Serpent's tAIL (1995)
20. Verfaille, V., Wanderley, M.M., Depalle, P.: Mapping strategies for gestural and adaptive control of digital audio effects. Journal of New Music Research 35(1), 71–93 (2006)
21. Wanderley, M., Battier, M. (eds.): Trends In Gestural Control of Music. Ircam - Centre Pompidou (2000)
22. Zadel, M.: Graphical Performance Software in Contexts: Explorations with Different Strokes. PhD thesis, McGill University, Montreal, Quebec, Canada (2012)