DeviceCycle: rapid and reusable prototyping of gestural interfaces, applied to audio browsing by similarity

Christian Frisson, Benoît Macq
Université catholique de Louvain, TELE Lab
Place du Levant, 2
B-1348 Louvain-la-Neuve
forename.surname@uclouvain.be

Stéphane Dupont, Xavier Siebert, Damien Tardieu, Thierry Dutoit
Université de Mons, TCTS and MathRo Labs
Boulevard Dolez, 31
B-7000 Mons
forename.surname@umons.ac.be

ABSTRACT

This paper presents the development of rapid and reusable gestural interface prototypes for navigation by similarity in an audio database and for sound manipulation, using the AudioCycle application. For this purpose, we propose and follow guidelines for rapid prototyping that we apply using the PureData visual programming environment. We have mainly developed three prototypes of manual control: one combining a 3D mouse and a jog wheel, a second featuring a force-feedback 3D mouse, and a third taking advantage of the multitouch trackpad. We discuss benefits and shortcomings we experienced while prototyping using this approach.

Keywords

Human-computer interaction, gestural interfaces, rapid prototyping, browsing by similarity, audio database

1. INTRODUCTION

Human-computer interaction gathers multiple, interdisciplinary fields of expertise: interaction design, software development, cognition, ergonomics, and so on... There is a trend among researchers in engineering sciences to put forward new technologies before finding applications for it, while designers would rather first understand user requirements before tailoring interfaces dedicated to their needs. In practise, it is convenient to use both methods side by side.

1.1 Our Approach

We started conducting research from an engineering point of view: trying to improve methods and algorithms for, in our case, hypermedia navigation or browsing by similarity in multimedia databases. This is an emerging field covering several application domains and use cases. Browsing in audio database by similarity is not yet a common practise, it aims at solving current limitations in sound search, retrieval and discovery; used in domains such as sound design, soundtrack composition, DJ/VJ'ing, electroacoustic music composition, personal audio library listening, among others. The knowledge and experience regarding interactive systems dedicated to these tasks vary amongst practitioners and disciplines. Addressing their various needs thus requires a disciplined methodology.

1.1.1 *General user-centered method, parallelized usability tasks*

We extended our initial approach by borrowing a methodology from the field of usability. The recent book by Bernsen and Dybkjaer [3] addresses this topic to a great extent. The knowledge acquired by gathering literature regarding specific use cases is highly increased by undertaking contextual inquiries, which consist in interviewing experts, possibly with questionnaires, so as to understand their habits and needs regarding their practises. Brainstorming with these experts to produce paper mockups of the desired user interfaces is cost-effective, especially when it prevents the unnecessary development of would-be inadequate prototypes. We have started investigating these axes. After collecting this information, several cycles of prototypes of user interfaces need to be designed, built, tested with users, and refined.

The time-consuming task of software and hardware development should be run in parallel with these usability tasks so as to ensure on-time delivery, and to ease the assembly of prototypes.

1.1.2 *Focus on rapid prototyping*

In this paper, we will focus on one single task pertaining to this approach: the rapid prototyping of software and hardware interfaces, following the design phase, but before running mandatory usability tests for validation of the prototypes.

Frameworks or toolboxes for rapid prototyping of multimodal applications should meet the following requirements:

- rapid: quicker to develop than finalized products;
- modular: proposing a component-based software and hardware architecture;
- realtime: seamless human-computer interaction with no perceivable latency;
- reusable: modules or components developed should fit well into new projects with little adaptation;
- understandable: visual representations of the underlying pipelines or mappings should be obvious;
- flexible: using libraries released under open licenses;
- generic: should work and fit well with most platforms and peripherals;
- sustainable: leaning towards the choice of environmental-friendly hardware.

Several frameworks, toolboxes and workbenches for rapid prototyping of multimodal interfaces have been proposed in the past, among which the following are still available and maintained: Amico[1], HephaisTK[2], OpenInterface[3][10], Squidy Lib[4]. Some of these are surveyed in [6]. However, we opted for dataflow environments, initially pushed forward by the signal processing communities, because simple interfacing is offered (audio and video input/output, gestural controllers and sensors interfacing, network communication) and because we had previously acquired knowledge and training in using these.

For the prototyping of user interfaces, we chose to use PureData[5], as it is an opensource (free), cross-platform, modular environment for visual programming. One convenient feature of PureData for interaction designers is the ability to edit mappings at runtime, on-the-fly, without recompiling. Jamoma[6] similarly simplifies the prototyping of dataflow applications, but inside the commercial Max/MSP environment, and less dedicated to multimodal interaction. Lawson's paper [10] compares the OpenInterface multimodal workbench with the PureData dataflow environment.

1.2 Our context of application: browsing audio loops by similarity and manipulation

1.2.1 AudioCycle, our test application

The AudioCycle application had already been presented [13, 7]: it offers an interactive visualization of a database of audio loops, with distances estimated from extracted signal-based features. A two-dimensional visualization has already been proposed, as illustrated in figure 1. This application has since then been extended to the MediaCycle framework, supporting other types of media such as images and videos. The framework relies on cross-platform opensource libraries, for instance for the AudioCycle application: OpenAL for the audio feedback, OSG for the OpenGL scene-based view, and Qt for the GUI.

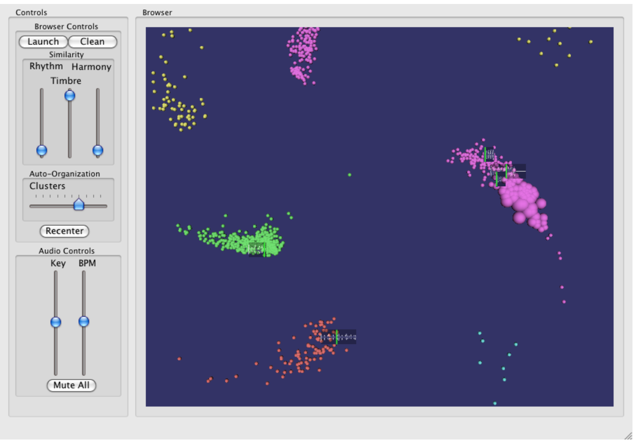

Figure 1: Screenshot of our test prototype for audio browsing by similarity: AudioCycle.

1.2.2 Other applications featuring dedicated gestural interfaces

A consistent survey of emerging applications for browsing audio databases by similarity, notably Musicream, MusicRainbow, SmartMusicKiosk and SoundSpotter, is available in [4]. The type of media in the audio database varies amongst these applications: samples, loops, and music libraries. As we are focusing on gestural control, we note that among these, MusicRainbow allows one to dial music genres for browsing a song database intuitively using a rotary controller. Alternatively, applications such as SongExplorer [9] have been proposed for interacting with sound content, for instance a music collection, using multitouch tables and/or "tangible" objects. In SoundTorch [8], the user can browse through a song library using a Wii remote controller.

Other inspiring dedicated controllers have been associated to audio manipulation applications, extending beyond browsing by similarity. Villar et al designed a low-cost system for DJ's using repurposed hard-drives [14]. Beamish et al [1] proposed force-feedback versions of the DJ elementary controllers: turntables and crossfaders. Chu [5] designed a force-feedback rotative knob for sound editing purposes.

1.2.3 Restriction to manual controllers

Expert users that would use a similarity-based approach to browse a collection of sounds for professional purposes require small movements considering their day-long sessions. We thus chose to address only manual controllers that can sit on top of desks or tables, requiring less bodily movements or aerial gestures. Relatedly, Dan Saffer opposes "touch" to "free-form" modalities in his taxonomy of gestural input modalities [11].

2. METHOD AND ACHIEVEMENTS

2.1 OpenSoundControl support

We have added OpenSoundControl (OSC) support to the MediaCycle framework with a dedicated namespace, so as to allow the external control of MediaCycle applications by more modalities, particularly gestural input.

We have identified two modes of interaction with AudioCycle: navigation (such as moving, hovering, zooming the database view) and manipulation (notably the playback of sound, with optional pitch-independent time stretching or beat-matching). For prospective collaborative uses of the application, using multiple control agents, we have chosen to specify numbers identifying the user and possible agents tied to the modes of interaction. For example, a single user using one browser to move the 2D view space may send: `/audiocycle/1/browser/1/move` <x><y>.

2.2 Discovering off-the-shelf devices

Before creating new dedicated gestural interfaces, we wanted to explore the limitations of off-the-shelf devices. We have investigated three categories of devices:

1. USB Human Interface Devices (HID), particularly Contour Design Shuttle jog wheels and 3dconnexion Space Navigator 3D mice, using the `[hidio]` object by Steiner et al [12], offering several improvements over `[hid]` (notably hotplugging devices);

2. force-feedback devices, notably the Novint Falcon 3DOF force-feedback mouse, operated with the `[np_nifalcon]` object using a reverse-engineering driver library[7], assorted with the HSP set of abstractions by Berdahl et al [2] facilitating the starting up with basic physical effects;

[1] http://amico.sourceforge.net
[2] http://sourceforge.net/projects/hephaistk/
[3] http://www.openinterface.org
[4] http://www.squidy-lib.de
[5] http://www.puredata.info
[6] http://www.jamoma.org
[7] http://sourceforge.net/projects/libnifalcon

3. multitouch trackpads, using the [fingerpinger] object that we ported from its initial Max/MSP implementation[8] to PureData, allowing to access information of blobs detected from fingers hovering the Apple Macbook Multitouch Trackpad, only available on Apple OSX currently.

For instant reusability of the controllers and associated objects, we created a set of abstractions following the same scheme for each (for instance a toggle connected at the left input of each enables to turn on or off the probing of the device), we ensured that the layout of the data visualization widgets on the patch remains coarsely equivalent regarding how related controls are laid out on the devices, as illustrated in figure 2.

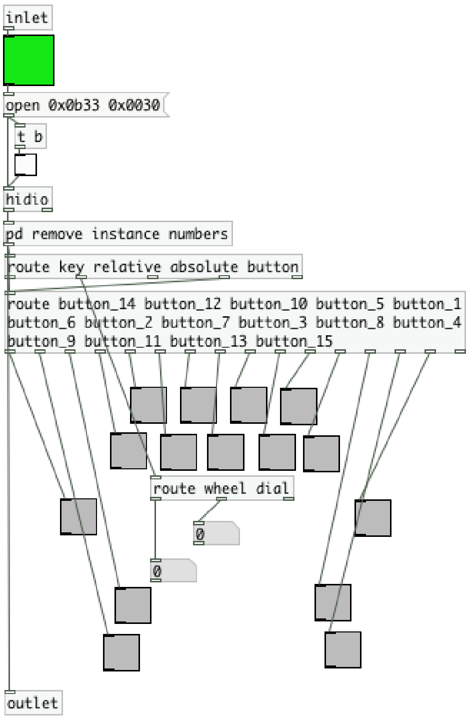

Figure 2: Abstraction for rapid interfacing of a Contour Design Shuttle Pro2 jog wheel.

For most USB HID devices, we had to uninstall the official drivers that circumvent implementation issues by converting raw events into mouse/keyboard events.

2.3 Rapid prototyping: producing mappings

We opted for three test prototypes:

1. bimanual control featuring a 3D mouse for the navigation in the audio database and a jog wheel for the manipulation of audio loops, is illustrated in figure 3;

2. a force-feedback version of the navigation, replacing the 3D mouse by a 3DOF force-feedback mouse;

3. a keyboard and multitouch trackpad combination

Figure 4 illustrates that mappings can be easily edited. Basic control widgets, such as toggles and sliders, are provided with the default PureData installation, and allow one to visualize values and test mappings offline. Objects and abstractions from the PureData extended release, such as

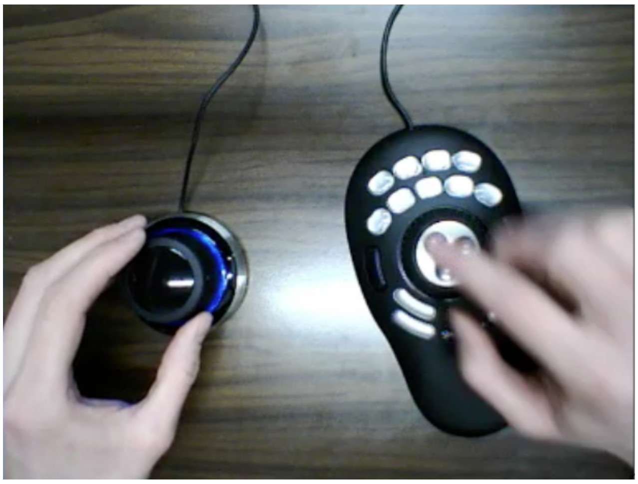

Figure 3: Bimanual audio browsing (left hand, 3D mouse) and manipulation (right hand, jog wheel) in action.

[scale] from the maxlib library, increase the speed of development and help reduce the visual clutter in the patch. For less straightforward event management, the [expr] object helps to define conditional sequences. OSC objects help to easily send filtered and conditioned events to the AudioCycle application, and to receive information for force-feedback.

3. DISCUSSION

3.1 Prototypes diversity

The first prototype reduces the level of movements down to wrists-only. As each device is assigned to only one interaction mode (navigation vs manipulation), the devices can be placed on a desk or table at convenient locations regarding hands so as to prevent long-term injuries.

Regarding the second prototype, even if usability settings are provided with the device, a suitable position is hard to find due to the size of the device, especially if coupled with another device. One strength of this prototype is the added value of force-feedback; we believe that it might increase the speed of finding a given target sound. For instance, as represented in figure 4, the friction activated when passing through loops might decrease the speed of reaching a given target, but increase the accuracy of positioning. A gravitational system analogy of attracting the force-feeback pointer towards barycenters of clusters can also be considered. Such systems can also benefit to users with disabilities, such as blind people.

For multi-purposed applications, the mouse and keyboard combination should remain usable as it is still the standard setup. This was the first supported user interface of the AudioCycle application. The third prototype augments this combination with a multitouch trackpad. This prototype can alternatively be used for prototyping small form factor multitouch applications.

3.2 Visual versus script-based prototyping

Some simple event mapping transformation were less straightforward than expected, for instance changing rotation from absolute to relative. This could have been more easily accomplished with script-based programming, which is more efficient for conditional and procedural statements, data recording at given time stamps (storing in a variable), and large datasets using dedicated types. OpenInterface ad-

[8] http://www.anyma.ch/2009/research/multitouch-external-for-maxmsp/

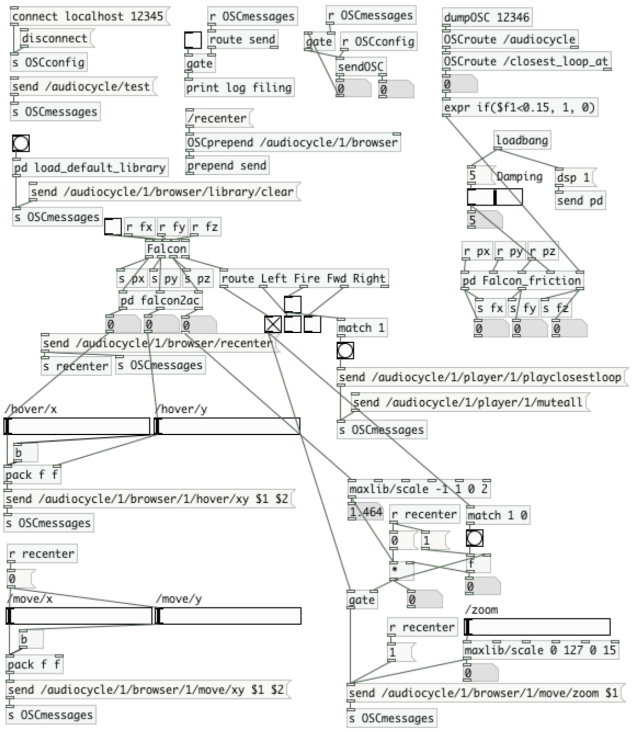

Figure 4: Rapid mapping example using a Novint Falcon force-feedback 3D mouse: samples are hovered with friction using the X and Y axes, zoom by the Z axis forced to its initial position.

dresses this issue by allowing the execution of Matlab scripts. Dataflow patches remain efficient for giving an overall summary of the executed pipeline.

3.3 Initial requirements assessment

Most of our initial requirements for rapid prototyping are met, with the following exceptions:

- visual programming prevents easy reusability between the prototype and the final product, as visual code needs to be converted into textual code;
- the GPL licence of PureData prevents from using it in commercial projects.

4. CONCLUSIONS AND PERSPECTIVES

We have set up a working environment for rapid and reusable prototyping of gestural interfaces. The next step consists in defining an usability protocol so as to determine which gestural interfaces provide the best results in a given application scenario with specific users and application domains. Since we have tested off-the-shelf components, we will try to repurpose some of these devices as in [14], for instance for the creation of a force-feedback jog wheel.

5. ACKNOWLEDGMENTS

All authors are supported by numediart[9], a long-term research program centered on Digital Media Arts, funded by Région Wallonne, Belgium (grant N°716631).

We wish to express our gratitude to our collaborators that contributed to the development of AudioCycle: Nicolas d'Alessandro, Thomas Dubuisson, Raphaël Sebbe, Jérôme Urbain.

[9] http://www.numediart.org

6. REFERENCES

[1] T. Beamish, K. van den Doel, K. MacLean, and S. Fels. D'groove: A haptic turntable for digital audio control. In *Proceedings of the 2003 International Conference on Auditory Display (ICAD)*, 2003.

[2] E. Berdahl, G. Niemeyer, and J. O. Smith III. HSP: A simple and effective open-source platform for implementing haptic musical instruments. In *Proceedings of NIME*, 2009.

[3] N. O. Bernsen and L. Dybkjaer. *Multimodal Usability.* Human-Computer Interaction Series. Springer, 2009.

[4] M. A. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney. Content-based music information retrieval: Current directions and future challenges. In *Proceedings of the IEEE*, volume 96, 4 2008.

[5] L. L. Chu. Using haptics for digital audio navigation. In *Proceedings of the International Computer Music Conference (ICMC)*, 2002.

[6] B. Dumas, D. Lalanne, and S. Oviatt. *Human Machine Interaction: Research Results of the MMI Program*, chapter Multimodal Interfaces: A Survey of Principles, Models and Frameworks, pages 3–26. Springer-Verlag, Berlin, Heidelberg, 2009.

[7] S. Dupont, T. Dubuisson, J. Urbain, C. Frisson, R. Sebbe, and N. d'Alessandro. Audiocycle: Browsing musical loop libraries. In *7th International Workshop on Content-Based Multimedia Indexing (CBMI)*, pages 73–80, Chania, Crete, June 3-5 2009.

[8] S. Heise, M. Hlatky, and J. Loviscach. Soundtorch: Quick browsing in large audio collections. In *125th Audio Engineering Society Convention*, number 7544, 2008.

[9] C. F. Julià and S. Jordà. Songexplorer: A tabletop application for exploring large collections of songs. In *10th International Society for Music Information Retrieval Conference (ISMIR 2009)*, 2009.

[10] J.-Y. L. Lawson, A.-A. Al-Akkad, J. Vanderdonckt, and B. Macq. An open source workbench for prototyping multimodal interactions based on off-the-shelf heterogeneous components. In *Proceedings of the 1st ACM SIGCHI symposium on Engineering interactive computing systems (EICS'09)*, 2009.

[11] D. Saffer. *Designing Gestural Interfaces.* O'Reilly Media, Inc., 2009.

[12] H.-C. Steiner, D. Merrill, and O. Matthes. A unified toolkit for accessing human interface devices in pure data and max/msp. In *Proceedings of the 2007 Conference on New Interfaces for Musical Expression (NIME07)*, 2007.

[13] J. Urbain, T. Dubuisson, S. Dupont, C. Frisson, R. Sebbe, and N. d'Alessandro. Audiocycle: A similarity-based visualization of musical libraries. In *IEEE International Conference on Multimedia and Expo (ICME)*, pages 1847–1848, Cancun, Mexico, June 28 - July 3 2009.

[14] N. Villar, M. Jervis, A. Lang, and H. Gellersen. The colordex dj system: A novel interface for music mixing. In *Proc. New Interfaces for Musical Expression (NIME'07)*, 2007.