

Improvised Interfaces for Real-Time Musical Applications

Jonathan Aceituno

INRIA Lille

Villeneuve d'Ascq, France

jonathan.aceituno@inria.fr

Julien Castet^{1,2}

¹University of Bordeaux, ²LaBRI, ³SCRIME, ⁴INRIA Bordeaux

351 cours de la Libération, 33405 Talence, France

castet@labri.fr

Myriam Desainte-Catherine^{1,2,3}

myriam@labri.fr

Martin Hachet^{4,1}

hachet@labri.fr

ABSTRACT

Computers offer a wealth of promises for real-time musical control. One of them is to enable musicians to change the structure of their instruments in the same time they are playing them, allowing them to adapt their tools to their wills and needs. Few interaction styles provide enough freedom to achieve this. Improvised interfaces are tangible interfaces made out of found objects and tailored by their users. We propose to take advantage of these improvised interfaces to turn the surrounding physical environment into a dynamic musical instrument with tremendous possibilities. Methods dealing with design issues are presented and an implementation of this novel approach is described.

Author Keywords

Tangible interaction, augmented reality, improvised interfaces, musical control, dynamic configuration, physical model.

ACM Classification Keywords

H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems. H.5.2 [Information Interfaces and Presentation]: User Interfaces. H.5.5 [Information Interfaces and Presentation]: Sound and Music Computing.

General Terms

Human Factors, Design.

INTRODUCTION

Musical performance is usually divided into two successive parts: preparation and playing. For example, a classical guitar player first prepares his instrument by ensuring its strings are in condition and properly tuned, and only then he plays the guitar by executing specific gestures on it. Such a sequential process is ubiquitous in traditional musical performance. Indeed, acoustic instruments are hardly modified and played at the same time, as many of them are two-handed.

Computer music introduced automation and looping into the workflow, allowing the musician to develop one-man

band abilities without much cognitive overload. What is controlled ranged from only one instrument to a complex structure of musical processes and related controllers. This brought the basic need to seamlessly rearrange the structures as the performance goes on, what we call *dynamic configuration*. Computer music tools should enable musicians to reorder the preparation-playing sequence and even to go back and forth between these phases. However, existing tools rarely offer such options in a satisfactory manner.

In this paper, we inquire into a special category of tangible interfaces that are tailored by the user at runtime using found objects in the surroundings to meet her requirements at the moment. We state that these improvised interfaces are suitable for music creation and ease dynamic configuration by allowing to build adapted musical structures and controllers on the fly. First, we define the term *improvised interface*, showing how it differs from similar ideas and how it contributes to dynamic configuration in a musical context. Second, we propose to relieve some weaknesses of improvised sensing systems for musical control with a physical model. Third, we emphasize the need for adapted interaction techniques and introduce two examples of them. Finally, we present design details for an improvised musical interface we have implemented and detail a use case illustrating the potential of our approach.

RELATED WORK

Interaction styles providing direct manipulation tend to spatially merge representation and control of musical processes, therefore allowing direct control over their structure[5]. As the same modalities enable performance or setup, they are more likely to be executed in no precise order and at any moment of the musical performance. Tangible user interfaces (TUI) allow such direct manipulation and dynamic configuration when involving a system of objects that can be added to or removed from a reference frame. For instance, the reacTable, a tabletop musical tangible instrument, provides a set of objects whose physical appearance differ according to their musical function (sine wave generator, echo effect, etc.). The structure of the instrument is changed simply by adding, removing or moving objects on the table[9].

Several properties of purposely designed object system TUIs restrict dynamic configuration. The number of possible musical structures one is allowed to build is determined by the physical bounds of the interface, the

number of possible objects involved and other space-related issues like physical clutter. On the contrary, tangible augmented reality (tangible AR) is not subject to such constraints and offers to visually augment physical objects and to let their physical manipulation be the primary means of interaction[2]. An early example of a tangible AR musical interface is Augmented Groove[10], where special cards representing short musical sequences could be manipulated, added or removed from the sight of the users, so as to change the undergoing musical structure. Opportunistic music[7] also explores the tangible AR setting and proposes to exploit physical surfaces of interesting found objects as controllers, but the interesting objects and the musical processes are defined beforehand. The potential of opportunistic music for dynamic configuration has yet to be harnessed.

IMPROVISED INTERFACES

We say that a tangible AR interface is *improvised* when the interaction is supported by non-specific and formerly unknown physical objects. This means that the physical part of these objects are not known by the designers, and their virtual part does not assume anything fixed about the physical part. The central implication of such interfaces is that a significant part of design decisions has to be delayed until runtime, so they are up to either the computer or the user. We think the second case is much more interesting because the user acts according to the current social or environmental context, her needs, intents, interpretations and personal experience. She might then select the most relevant physical object to embody chosen digital functions at a precise moment. Moreover, in case of unintended use, improvised interfaces make it possible to pick any physical object rather than allowing to choose between a set of predefined and possibly inadequate entities.

Similar ideas

Several existing systems partly illustrate this concept. For instance, MemoICON is an interactive tabletop tactile display that allows to use physical everyday objects as icons for particular digital contents or actions[3]. A coffee cup is put on the table, then a user drags data shown on the table to the cup's vicinity with her fingers in order to associate them. This system is partially improvised, because although non-specific objects can be used, the tactile display remains the primary channel of interaction.

The notion of improvised interfaces exists to some extent in Opportunistic Controls[8] (OCs), which are 3D widgets coupled to physical objects present in the environment but otherwise unused, like button-looking bolts or slider-looking pipes. Those widgets are designed in advance and can be activated with gestures on their physical counterparts. Although the principle behind OCs seem close to improvised interfaces, the latter term is less general because apart from being opportunistic, i.e. making use of non-specific physical objects to support man-machine interaction, improvised interfaces have no

former knowledge of the physical objects that will be employed.

Dynamic configuration

The dynamic binding of physical and digital parts and the availability of any existing physical object for inclusion in the interactive context are characteristic properties of improvised interfaces. Their combination provides a high degree of adaptability and openness that encourages creativity. This is particularly interesting for problem solving or artistic applications. Dynamically designing new objects that contribute to the musical structure is a standard procedure in improvised interfaces for musical creation and performance, as tangible objects embody musical processes that can be directly controlled by physical manipulation. These interfaces enable dynamic configuration by nature.

METHODS

The improvised nature of the interactions described above has several effects on the design of specific sensing systems. For the same reasons as in AR systems, computer vision might be the sensor of choice. However, its adequacy for musical gestures is far from being obvious. In this section, we suggest a way to mitigate this problem and propose interaction techniques for dynamic configuration in this context.

Virtual coupling

Every sensor-based interface has to operate under low latency and jitter in order to ensure coherence and causality for the user[1]. Interactive musical performance systems need even tighter requirements. Wessel and Wright state that the time between a gesture and its computer generated audible reaction should be below 10 milliseconds[11]. Improvised interfaces are more likely to rely on computer vision, however such techniques do not suit the needs for musical performance very well. Consumer-grade cameras hardly operate above 60 Hz. Computationally expensive vision algorithms can increase the latency of the whole system. External conditions such as luminosity or motion blur decrease their recognition rate. These factors make it hard to correctly interpret moderately fast and continuous gestures and to keep the interaction unambiguous.

We propose to attenuate these problems by introducing virtual coupling, a physical model simulating an indirection between physical objects being sensed by the computer and their visual and sound augmentations. We associate a *ghost* to each physical object that is active and tracked by the computer so that they seem physically bound with an elastic link. The ghost object is shown to the user as a visual augmentation and follows the real object with delay and a smooth motion that is generated at high frequencies. In further processings, the system does not interpret directly the physical object but the ghost object's motion. We drew our inspiration from similar methods that have been employed in haptics to guarantee the passivity of haptic renderers[4]. As this effect is clearly visible, chances

are that the user can anticipate the motion of the ghost object based on the physical object's motion. Causality and coherence is guaranteed and continuous motions, if not too fast, are correctly rendered even if they are poorly sampled by the sensors. We suspect that by artificially showing a steadier delay in such a natural way, vision-based improvised interfaces would appear less disturbing.

An object and its ghost are modeled by the point particles \hat{O} and O respectively. The location of \hat{O} in space is $(x_{\hat{O}}^1, x_{\hat{O}}^2, x_{\hat{O}}^3)$ and it is determined entirely by the sensing system. The particles are bound together with a damper, of damping coefficient z , and a spring, of spring constant k , in parallel. The forces acting on O from \hat{O} are then the sum of the viscous and elastic forces. According to the second law of Newton, for $i = 1, 2, 3$:

$$\frac{d^2 x_O^i}{dt^2} = k(x_{\hat{O}}^i - x_O^i) - z\left(\frac{dx_{\hat{O}}^i}{dt} - \frac{dx_O^i}{dt}\right) \quad (1)$$

This model is simulated in discrete time. Let $(\hat{x}_n^1, \hat{x}_n^2, \hat{x}_n^3)$ be the location of \hat{O} in space, and (x_n^1, x_n^2, x_n^3) be the location of O at time n . For $i = 1, 2, 3$, the sum of all forces acting on O is:

$$\sum F_n^i = k(\hat{x}_n^i - x_n^i) - z(\hat{x}_n^i - \hat{x}_{n-1}^i - (x_n^i - x_{n-1}^i))$$

According to (1), the new position of the ghost object O can be calculated at time $n + 1$ using previous positions and the forces at instant n :

$$x_{n+1}^i = 2x_n^i - x_{n-1}^i + \sum F_n^i$$

The parameters k and z and the frequency of the simulation have to be fixed manually. We found $k = 0.08$ and $z = 0.05$ to be acceptable at 100 Hz.

Interaction techniques

Interaction with improvised interfaces is driven by creativity and experimentation, but the freedom it offers comes at the expense of a much higher entry threshold. A challenge of improvised tangible AR interfaces is to provide straightforward interaction techniques for handling the digital augmentations of the physical objects. The user must be able to involve found objects into the interaction context, and to couple them at will with new or existing digital objects. Because of the dynamic nature of the possible couplings, we found the menu metaphor to be adapted to these requirements and investigated two menu-based interaction techniques: alpha object and shake menus. For both techniques, a user can take any physical object that has not been previously used in the system and then invoke a menu, in order to select and associate a digital counterpart.

Our first attempt was to designate a special object, the alpha object, responsible for any dynamic configuration task. The user places the alpha object over another physical object. After a moment, a radial menu appears and the user moves the alpha object out of the object underneath. The

first menu item is then highlighted and the user can select any menu item by rotating the alpha object. She moves the alpha object back over the other object in order to confirm her choice.

We also tried to adapt shake menus, which are radial menus centered on a particular physical object and invoked after it has been shaken, in the context of tangible AR. We implemented a variant of the display-referenced placement[12]. After the user has shaken the object long enough, the menu doesn't follow it anymore and the user can select a menu item by aligning the object with it. The selected menu item is highlighted as long as another menu item is not selected. The user has to shake the object a second time to confirm her selection.

We found both approaches to be globally satisfying. They are adapted to collaborative uses, as multiple alpha objects or shake menus can be used at once. However, selecting menu items by rotating the alpha object is laborious, and the shaking gesture is not adapted to large objects with restricted degrees of freedom. A hybrid strategy might be more suitable.

EXPERIMENTING WITH AN IMPROVISED MUSIC ENVIRONMENT

We wanted to experiment the previous ideas within an improvised music environment serving as a base for future research and developments. After a concise description of how the system is implemented, we explain how dynamic configuration is represented and detail an example of use.

Implementation

The system is written in C++ and requires headphones, a camera and a display, either a projector or a HMD. The camera captures the image of objects tagged with fiducial markers whose position in space is determined thanks to the ARToolkitPlus library[6]. After further processing, sound is produced through the headphones and a visual feedback is shown through the display.

We chose to model the musical processes using the modular synthesis paradigm. Each process is either a source, able to generate signal, or an effect, able to transform signal passing through it. Audio signal flows from sources, and eventually through effects, to a special process called the master output, which represents the headphones.

Behaviors

We defined a simple framework describing the relationships between the physical environment and the virtual environment. On one side, the physical environment is the set of sensible physical entities, or objects, that are modeled by the interface, based on its sensing systems. On the other side, the virtual environment is the set of musical processes that are meant to be controlled. A behavior class is a canonical reaction in both environments to variations of an unknown object in the physical environment model.



Figure 1. The user attaches a behavior to a wind-up toy robot (a) in order to build a mechanical sequencer (b).

A behavior is an instance of a behavior class that is attached to a particular and known physical object. The set of all behaviors define the possible interactions at a time.

The core of the system is not more than a dynamic behavior coupling interface and the specialisation of available behaviors for musical control make it a dynamic configuration enabled instrument. The set of available behavior classes defines how to perform with the musical interface, so their design must follow a consistent strategy. Moreover, behavior classes must allow expressive control. We didn't want to interfere with these problems at this time and focused instead on getting a first functional preview of a musical improvised interface.

Use case: mechanical sequencer

Several basic musical behavior classes were implemented. Among them are the guitar string and the playback head. This is enough to build a mechanical sequencer using physical objects found in a child's room (see Figure 1). The user stumbles upon a deck of cards and lays four of them horizontally on a table. She takes one card and shakes it until a menu appears. She shakes the card a second time after highlighting the guitar string menu item, in order to confirm that the card will behave like a guitar string. Every card on the table is associated with a guitar string behavior in a similar fashion. A playback head behavior is then attached to a wind-up toy robot placed on the left and facing the right of the table. While the robot moves forward, every augmented object perpendicular to its trajectory is triggered. As the robot continues its path, the user adds two more cards-strings on the left. At the end, she handles the robot and strums the strings by moving it across the table: the sequencer is also a guitar.

CONCLUSION

We have presented improvised interfaces, a particular case of tangible AR interfaces where any physical object can dynamically participate to the interactive context, and suggested that they are adapted to real-time musical applications. They leverage musical creativity by giving the user a designer role and by facilitating dynamic configuration. As computer vision is the likeliest alternative for sensing in improvised interfaces, we proposed to ease its drawbacks for musical control by introducing a physical model called virtual coupling and

we conjectured its suitability for some musical gestures. We discussed the need for interaction techniques adapted to improvised interfaces and introduced two menu-based examples. We implemented an improvised music environment following our approach and methods. As an example, we detailed the creation of a mechanical sequencer out of found objects.

We hope this preliminary work will inspire further research on improvised interfaces. Deeper studies are required to validate our hypotheses on virtual coupling and interaction techniques. The drastic sensing conditions combined to the tight requirements of musical control make the design of such interfaces challenging. The consequences for interaction of the users' extreme involvement on interface design must be investigated. We also think that other fields of application can benefit from improvised interfaces. The perspective of mobile, persistent and networked improvised environments drives interesting questions too.

REFERENCES

1. S. Antifakos, J. Borchers, and B. Schiele. Designing physical interaction with sensor drawbacks in mind. In *Proc. PI03*, page 56. Citeseer, 2003.
2. M. Billinghamurst, H. Kato, and I. Poupyrev. Tangible augmented reality. In *Proc. ACM SIGGRAPH ASIA 2008*, pages 1–10. ACM, 2008.
3. B. Chen, K. Cheng, H. Chu, S. Kuo, R. Liang, M. Yu, R. Liang, H. Lin, and Y. Chu. MemoICON: using everyday objects as physical icons. In *Proc. ACM SIGGRAPH ASIA 2009*, pages 78–78. ACM, 2009.
4. J. Colgate, M. Stanley, and J. Brown. Issues in the haptic display of tool use. In *Proc. IEEE IROS*, volume 3, pages 140–145. IEEE, 1995.
5. J.-M. Couturier. A model for graphical interaction applied to gestural control of sound. In *Proc. SMC*, 2006.
6. W. Daniel and S. Dieter. ARToolKitPlus for pose tracking on mobile devices. In *Proc. CVWW*, 2007.
7. M. Hachet, A. Kian, F. Berthaut, J.-S. Franco, and M. Desainte-Catherine. Opportunistic music. In *Proc. JVRC (EGVE - ICAT - EuroVR)*, pages 45–51, 2009.
8. S. Henderson and S. Feiner. Opportunistic tangible user interfaces for augmented reality. *IEEE Transactions on Visualization and Computer Graphics*, 16(1):4–16, 2010.
9. M. Kaltenbrunner, G. Geiger, and S. Jordà. Dynamic patches for live musical performance. In *Proc. NIME*, pages 19–22, 2004.
10. I. Poupyrev, R. Berry, J. Kurumisawa, K. Nakao, M. Billinghamurst, C. Airola, H. Kato, T. Yonezawa, and L. Baldwin. Augmented groove: Collaborative jamming in augmented reality. In *Proc. ACM SIGGRAPH Conference Abstracts and Applications*, page 77, 2000.
11. D. Wessel and M. Wright. Problems and prospects for intimate musical control of computers. *Computer Music Journal*, 26(3):11–22, 2002.
12. S. White, D. Feng, and S. Feiner. Interaction and presentation techniques for shake menus in tangible augmented reality. In *Proc. IEEE ISMAR*, pages 39–48, 2009.