Pointing Fingers: Using Multiple Direct Interactions with Visual Objects to Perform Music

Jean-Michel Couturier
LMA-CNRS
31, chemin Joseph Aiguier
13402 Marseille Cedex 20, France
0033 491 16 42 10
couturier@lma.cnrs-mrs.fr

Daniel Arfib
LMA-CNRS
31, chemin Joseph Aiguier
13402 Marseille Cedex 20, France
0033 491 16 42 10
arfib@lma.cnrs-mrs.fr

ABSTRACT
In this paper, we describe a new interface for musical performance, using the interaction with a graphical user interface in a powerful manner: the user directly touches a screen where graphical objects are displayed and can use several fingers simultaneously to interact with the objects. The concept of this interface is based on the superposition of the gesture spatial place and the visual feedback spatial place; it gives the impression that the graphical objects are real. This concept enables a huge freedom in designing interfaces. The gesture device we have created gives the position of four fingertips using 3D sensors and the data is performed in the Max/MSP environment. We have realized two practical examples of musical use of such a device, using Photosonic Synthesis and Scanned Synthesis.

Keywords
HCI, touch screen, multimodality, mapping, direct interaction, gesture devices, bimanual interaction, two-handed, Max/MSP.

1. INTRODUCTION
In computer music, different strategies are possible to control sound processes. A first one consists in using the computer properties of calculation power and flexibility in the design phase of an instrument. Today, many researches are conducted to create powerful digital musical instruments. To design them, a critical part of the work consists in the mapping between the gestural devices and the sound processes to control [1]. Those instruments tend to reproduce the "instrumental link" [13] that is intrinsic to the acoustic instruments and that has often disappeared in the electronic and numerical systems.

A second strategy consists in using the computer for its powerful interaction trough a graphical user interface (GUI). Regarding today musical softwares, they essentially use a mouse and a keyboard with a current GUI: all sound parameters are controllable via graphical objects that generally represent real objects like piano keyboards, faders, etc. Complete studios equipments and electronic instruments emulators are now integrated in the computer. The GUIs tend to reproduce on the screen an interaction area close to the real one, like front panels of electronic instruments. The aim of such interfaces is to give the user the impression of real objects in front of him. Nevertheless, with a single mouse, the interaction process is poor: the gesture space (the place where is the mouse) is separated from the interaction space (the screen) and only one object can be manipulated at one time. This explains why many software programs are configured to use "external" devices like MIDI controllers, software-specific control surfaces or alternative controllers. In this case, the full system is similar to those of the first strategy; the graphical objects, which are designed for interaction, are only used for visual feedback or not used at all.

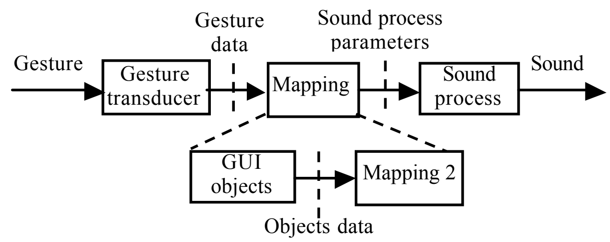

Figure 1. Mapping chain in a digital musical instrument. The first strategy links gesture data to sound parameters; the second strategy adds an additional step in the mapping: the graphical objects are linked to sound parameters and the gesture device can control any graphical object.

The system we introduce in this article enables the control of graphical objects in GUI's like real objects and rather follows the second strategy. This new powerful multimodal system, the *Pointing Fingers*, performs a direct control on GUIs with a multi-touch touchscreen-like device, designed for musical control. Section 2 introduces the interaction principle; section 3 describes the gesture device and section 4 the software implementation. Finally, in section 5, some musical examples of what is possible with such a system are exposed.

2. A NEW APPROACH IN INTERACTIVE SYSTEMS
The system is based on the combination of two crucial features: the superposition of both gesture spatial place and visual feedback spatial place and the ability to have multiple simultaneous controls when using a GUI. Some systems that have these two features already exist; one of them was developed to control musical processes: the Audio Pad [10], based on tangible interfaces [7] in which the objects to manipulate are real and interact with graphics. Our system is closer from current GUIs because the objects to manipulate are the virtual graphical objects displayed on screen.

This type of system provides the most direct and intuitive interaction possible: our fingers are manipulating graphical objects as if they were real objects. There are no material constraints on the objects: they can change in position, size, shape and function. It is possible to display some information beside the objects to help the user. It is a very efficient system to control virtual copies of real objects. Finally, interaction situations that are impossible in the real world can be implemented here, like manipulating moving objects, as it will be demonstrated in the section 5.

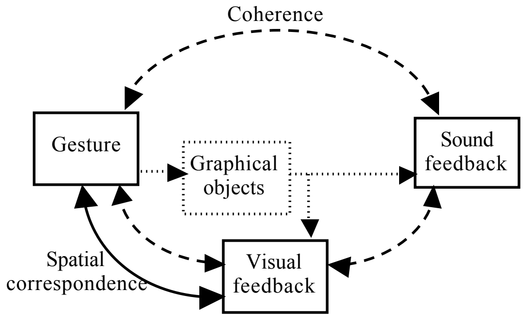

Figure 2. In this system with multimodal feedbacks, the relationship between action and perception is coherent; this coherence is reinforced by a spatial superposition of the gesture area and the visual feedback area. Those features add strong presence to the system.

In interaction with a real object, this object provides some haptic feedback: the contact with the object shape, the force it needs to be manipulated, the degrees of freedom it offers and the spatial limits of its displacement. This feedback is so important that the user could manipulate an object with the eyes closed. With our system, the haptic feedback is reduced to the contact between fingers and screen. Sight and hearing are fully used; sight permits to locate the position of the objects in the screen and hearing can reinforce sight when an object is manipulated, through the effect of manipulation on the sound.

The GUI of our system is close to those using a mouse to control graphical object; the differences are that the object needs a bigger size, because a fingertip is bigger than a mouse pointer. The screen area contains different interaction zones; each zone will have its own interaction mode and connection to the sound process parameters.

Different types of gestures are necessary to act in a zone: selection gesture to select the chosen zone among several zones, modulation or continuous gesture to modify the parameters that are associated with the zone, and decision gesture to stop the interaction. For example, if the user wants to manipulate the graphical object " fader", he selects this fader with one of his fingers, manipulates it, and then he lifts his finger off the screen area.

3. THE POINTING FINGERS SYSTEM

We want a device that follows our requirements: having multi-touches and interacting directly with the interface. Commercial touchscreens fulfill the second point, but unfortunately, they does not allow multi-touches. Many other solutions have been developed in different labs, as the following examples: the SmartSkin [11] system combines a prototype of multi-touch surface with a video projection; the vision-based finger tracking [6] determinates the fingers' positions through a video analysis; Mulder's system combines two CyberGloves and two Polhemus position/orientation sensors enables to find the position of the fingertips [9]. The device we have developed represents a simple alternative to all that exists.

3.1 The Gesture Device

The device we introduce now is a first prototype we have made to perform multi-touches on a screen. It consists of 2 semi-gloves (recovering the thumb and the index) with two 3D position/orientation sensors and two switches per hand (see Figure 3). This device is close to Mulder's CyberGloves and Polhemus system [9], but is less expensive in hardware and simpler to implement.

Figure 3. The gesture device uses *flock of birds* sensors with 4 *birds* (receivers). The *birds* are fixed on the thumb and the index of each hand so that no motion is possible between the *birds* and the fingertips. Switches are fixed on each fingertip and can be used like a mouse click button.

This device can give the position of 4 digits (the thumb and the index of each hand) with approximately 1 mm accuracy and the on/off values of the switches (an equivalent of the mouse click button) localized at the extremity of the fingers; those switch buttons indicate if the fingertips are physically touching the screen or not. All the data of the sensors are processed in the Max/MSP environment. The *flock of birds* [3] is a commercial device composed of a transmitter and several receivers, called *birds*; the device communicates with the computer trough a serial interface and a serial/USB converter. We use the *serial* object of Max to receive the data. The switches are connected to the electronic of an USB joystick and we receive its data, using the *insprock* object.

However, this device has some limitations. The *flock of birds* device introduces some latency: we have not measured it but we estimate it to be approximately 30 ms with four sensors; this lag is too important to create really reactive instruments, but is acceptable for our experiments and applications with modulation-like instruments. Another problem is the choice of the screen: CRT screens are disturbed by magnetic fields, and some LCD screens disturb the magnetic field of the sensor.

3.2 Converting the Data of the 3D Sensors

We have developed a specific C object for Max to transform the data of the *birds* and find the fingertips coordinates in the screen base. The sensor gives the absolute position and orientation of the 4 *birds* in space, relatively to the transmitter; with this data, the object calculates the position of the tips using the rotation matrix between each *bird* base and the transmitter base. This coordinates are then rotated and translated to the screen base and rescaled in order to obtain the position of the tips in pixel, which is the mouse coordinates unit (Figure 4). A calibration procedure calculates the screen position and size in the transmitter base and then determines the screen base.

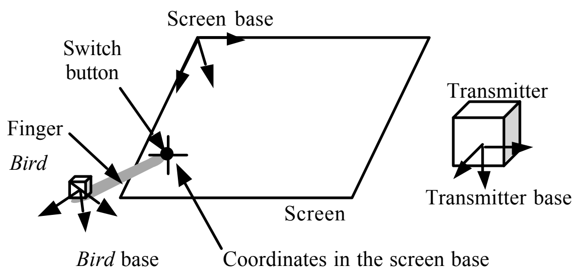

Figure 4. For one finger: the fingertip position is known in the *bird* base, and the *flock of birds* gives the *bird* relative-to-transmitter position/orientation. With the screen position, the program calculates the coordinates of the fingertip in the screen basis.

The object returns the (X,Y) coordinates of the four fingertips in the screen base. The switch button state, given by the *insprock* object, is added to the corresponding list.

4. CONTROLLING GRAPHICAL OBJECTS

In this section, we develop how the data of our gesture device or any equivalent device will be processed. Indeed, in our approach, we try to build modular systems. So the control of the graphical object is completely independent from the gesture device: we consider that any gesture devices that can give us lists with the point number, (X,Y) coordinates in the screen basis and the value of a on/off button can be used instead of the Pointing Fingers. For this reason, we will call *pointer* a point on the screen that is given by the gesture device. Our gesture device gives simultaneously 4 pointers.

We used the Max/MSP environment and we created a specific Max object to manage the data for a given zone of the screen: multipoint provides some confusion problems that did not exist with the only mouse. The object receives all data lists from all pointers. The delimitations of the object action zone are given by sending specific instructions to it. Figure 5 describes how this Max object manages multiple points for a given zone.

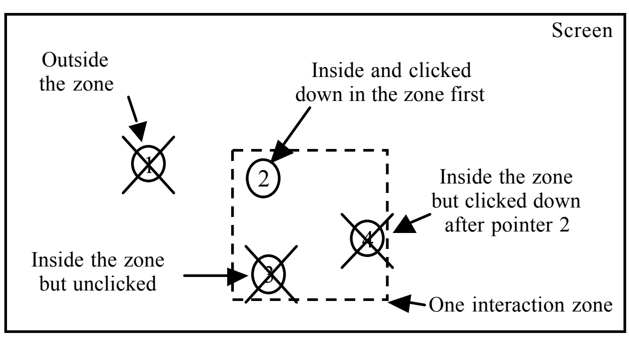

Figure 5. The Max object returns the coordinates of a pointer only if this pointer is clicked down in the zone of the object and if the zone does not contain another active pointer.

The outputs of this max object can be connected with many graphical objects, taking care of the coherence between the visual effects of the interaction on the graphical object and the position of the pointer on the screen.

This implementation is simple but sufficient to perform numerous things. Firstly, lots of Max graphical objects can be used with our Max object and can be manipulated simultaneously. Secondly, many original graphical objects or interaction zones can be created and used with our system, as the section 5 will show, and we can imagine multipoint interaction zones using several units of our Max object.

5. EXAMPLES

5.1 Control of the Photosonic Synthesis with the Pointing Fingers Device

Created by Jacques Dudon, the Photosonic instrument is an optical musical instrument based on the following principle: a solar photocell receives the rays of a light that are intercepted by a rotating disk and an optical filter (see figure 6). The electric current of the photocell is the audio signal that the instrument produces. An optical comb-filter can be placed on the trajectory of the light to modify the sound of the disk. The sound of the instrument depends on the position of the light, the waves inscribed on the photosonic disk, and the position of the filter.

We have made an emulation of the instrument in the Max/MSP environment, which was presented in NIME 2002 [2]. This digital version of the instrument uses a graphical tablet with a mouse and a pencil to control the position of the light and the filter. Now we will introduce an implementation in which the Photosonic synthesis is controlled by the *Pointing Fingers*.

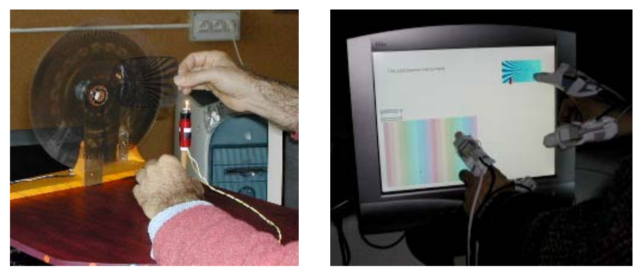

Figure 6. The optical instrument and its digital emulator's interface. The emulator proposes the same control than the real instrument, where the user moves a light in front of the disk and a filter between the disk and the photocell.

We have created a graphical interface close to the optical version of the instrument. This interface contains two interaction objects, like the real instrument: the light and the filter. The filter is the same than the optical one, with a rectangular shape. A circle represents the light and can be displaced below the disk, which is represented in a rectangle divided in several parts corresponding to the rings of the Photosonic disk. The interaction principle is here similar to the interaction with real objects: when one object (the light or the filter) is *clickdown* or selected on the screen, it follows the displacements of the fingertip. When the finger switch is *clickup*, the activation zone is the new position of the object. This interaction mode provides a digital instrument that is really close to the real optical instrument in terms of handiness.

5.2 String Control in Scanned Synthesis

Scanned Synthesis was developed by Verplank, Shaw and Mathews [12] and enables the generation of sound thanks to the slow movements of mechanical systems, which shape is used to create dynamical wavetables. We have implemented this technique with a circular string model in finite differences for the mechanical system. A C object that provides a high level control of the Scanned Synthesis was created [5] for the Max/MSP software and was used for the realization of a complete musical instrument demonstrated at NIME 2002 [4]. In these previous works, the string was put in motion by forces or by throwing it from a pre-definite shape. The string shape was only displayed as graphical feedback. However, The link between sound and the string visual representation is direct: the motion of the string is perceptible with the eyes and its speed corresponds to the speed of the human body gestures. Because of these features, we felt like interacting with the string directly using our fingers, in real time. Scanned Synthesis is the perfect synthesis technique to use with our new gesture device.

We then modified our C object in order to enable direct interaction with the string and we create a GUI that displays the string shape, an area to control the pitch (that was implemented in our instrument and controlled by a graphical tablet) and other controls.

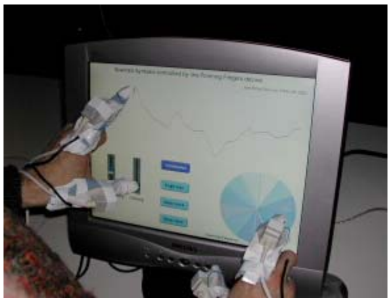

Figure 8. Direct manipulation of the string of the Scanned Synthesis algorithm. One can simultaneously interact with the string, control the pitch and change string parameters.

On the string, the interaction principle is the following: a finger makes a selection gesture up or down the string; according to its initial position, the finger pushes the string up or down. The pitch control is localized in another area and uses the angular frequency control developed by Kessous [8]. Two sliders are modifying the string damping and stiffness; 4 buttons enables to stop the sound and to choose to play a single note or chords. The sliders and the buttons are standard Max graphical objects that receive data of our system.

Scanned Synthesis is a complex method that disposes of a high number of parameters. Nevertheless, controlled by our system, this synthesis technique becomes easy to use and gives remarkable presence to the interaction.

6. CONCLUSION - PERSPECTIVES

The computer often seems to be a powerful creature inside a closed box. Its screen shows us marvelous worlds, but interacting with a mouse is frustrating, especially when we want to perform music. As the two examples shows, our system will help to design musical instruments that benefits of the advantages of the computers' universality and flexibility, through a powerful control of Graphical User Interfaces.

Our works on this system have just established its basis; in the future, we will develop new objects, implement other synthesis techniques and improve the system to provide a complete environment to create new digital musical instruments.

7. REFERENCES

[1] D. Arfib, J.M. Couturier, L. Kessous, V. Verfaille, "Mapping strategies between gesture control parameters and synthesis models parameters using perceptual spaces", Organised Sound 7(2), Cambridge University Press, pp. 135-152, 2002.

[2] D.Arfib, J. Dudon: "A digital emulator of the photosonic instrument", Proceedings of the 2002 Conference on New Instruments for Musical Expression (NIME-02), Dublin, Ireland, May 24-26, 2002.

[3] Ascension Technology Corporation, http://www.ascension-tech.com/

[4] J.M. Couturier, "A scanned synthesis virtual instrument", Proceedings of the 2002 Conference on New Instruments for Musical Expression (NIME-02), Dublin, Ireland, May 24-26, 2002.

[5] JM. Couturier, Scansynth~: Scanned Synthesis object for Max/MSP, 2002. http://www.lma.cnrs-mrs.fr/~IM/en_telecharger.htm

[6] C. von Hardenberg, F. Bérard, "Bare-Hand Human-Computer Interaction" Proceedings of the ACM Workshop on Perceptive User Interfaces, Orlando, Florida, USA, Nov. 15-16, 2001.

[7] H. Ishii and B. Ullmer, "Tangible Bits: Towards Seamless Interfaces between People, Bits, and Atoms", Proceedings of Conference on Human Factors in Computing Systems (CHI '97), ACM Press, pp.234-241, 1997.

[8] L.Kessous, "A two-handed controller with angular fundamental frequency control and sound color navigation", Proceedings of the 2002 Conference on New Instruments for Musical Expression (NIME-02), Dublin, Ireland, May 24-26, 2002.

[9] A. Mulder, S. Fels and K. Mase, "Design of Virtual 3D Instruments for Musical Interaction", Proceedings of Graphics Interface '99, pp. 76-83, Toronto, Canada, 1999.

[10] J. Patten, B. Retch, H. Ishii, "Audiopad: A Tag-based Interface for Musical Performance", Proceedings of the 2002 Conference on New Instruments for Musical Expression (NIME-02), Dublin, Ireland, May 24-26, 2002.

[11] J. Rekimoto, "SmartSkin: An Infrastructure for Freehand Manipulation on Interactive Surfaces", Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 113-120, Minneapolis, USA, 2002.

[12] B.Verplank, M. Mathews, R. Shaw, "Scanned Synthesis", "Proceedings of the 2000 International Computer Music Conference", p: 368-371, Berlin, Zannos editor, ICMA, 2000.

[13] M. Wanderley, "Performer-Instrument Interaction: Applications to Gestural Control of Music", PhD Thesis, Paris, University Pierre et Marie Curie - Paris VI, 2001. http://www.ircam.fr/wanderle/Thesis/Thesis_comp.pdf