

Piivert: Percussion-based Interaction for Immersive Virtual EnviRonmenTs

Florent Berthaut, Martin Hachet, Myriam Desainte-Catherine

▶ **To cite this version:**

Florent Berthaut, Martin Hachet, Myriam Desainte-Catherine. Piivert: Percussion-based Interaction for Immersive Virtual EnviRonmenTs. Symposium on 3D User Interfaces, Mar 2010, United States. p 15-18. hal-00522637

HAL Id: hal-00522637
https://hal.science/hal-00522637

Submitted on 1 Oct 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Piivert: Percussion-based Interaction for Immersive Virtual EnviRonmenTs

Florent Berthaut* Martin Hachet† Myriam Desainte-Catherine‡
Université de Bordeaux - LaBRI INRIA - LaBRI Université de Bordeaux - LaBRI

ABSTRACT

3D graphical interaction offers a large amount of possibilities for musical applications. However it also carries several limitations that prevent it from being used as an efficient musical instrument. For example, input devices for 3D interaction or new gaming devices are usually based on 3 or 6 degrees-of-freedom tracking combined with push-buttons or joysticks. While buttons and joysticks do not provide good resolution for musical gestures, graphical interaction using tracking may bring enough expressivity but is weakened by accuracy and haptic feedback problems. Moreover, interaction based solely on tracking limit the possibilities brought by graphical interfaces in terms of musical gestures.
We propose a new approach that separates the input modalities according to traditional musical gestures. This allows to combine the possibilities of graphical interaction as selection and modulation gestures with the accuracy and the expressivity of musical interaction as excitation gestures. We implement this approach with a new input device, Piivert, which combines 6DOF tracking and pressure detection. We describe associated interaction techniques and show how this new device can be valuable for immersive musical applications.

1 INTRODUCTION

Graphical musical interfaces have several advantages over hardware controllers. They may give more meaningful visual clues about musical processes as stated by Jordà [6], for example by displaying perceptual parameters, processes' status, or even musical structures. 3D environments add even more possibilities in terms of navigation [11], visualization, manipulation [7] [8], and immersion. In the case of multi-processes instruments, i.e instruments that handle several sound synthesis processes, this allows musicians to easily visualize and select the processes, represented by graphical objects, they want to control. One can navigate a 3D musical piece, and manipulate these audiovisual objects with a ray-casting metaphore, i.e a virtual ray going from an input device that can be aimed at 3D objects in the virtual environment.

Handheld input devices used to interact in these environments, i.e to control virtual rays, mostly rely on 3 or 6 degrees-of-freedom tracking often combined with push-buttons and joysticks. This is also the case for new gaming devices such as the Nintendo Wii remote-controller. Push-buttons or joysticks of these devices have lower resolutions than video, magnetic, or accelerometer based tracking. Thus they do not fit most musical gestures and reduce musical possibilities. For example, a push-button only outputs 0 or 1, so it cannot reproduce the dynamics of a percussion gesture. On the other hand, tracking-based graphical interaction provides sufficiently high resolution but one can not rely exclusively on it for musical interaction because of tracking accuracy problems and/or lack of haptic feedback, and interaction limitations. These limitations may explain why existing immersive instruments have not been much developed and played until now.

To overcome these issues, we propose to take inspiration from traditional musical instruments and to split 3D interaction with immersive instruments into several musical gestures, as explained in section 2. This allows us to combine advanced musical interaction, such as the applications described in [5] [14], with 3D graphical interaction. To explore this idea, we designed a new input device, called Piivert and described in section 3, which is based on 6DOF tracking and fine pressure sensing. Basically, with such a device, the musician will graphically select and manipulate audiovisual objects through a virtual ray metaphor, while using the pressure capabilities of the device to interact with the selected objects within fine musical controls. These interation techniques are presented in section 4.

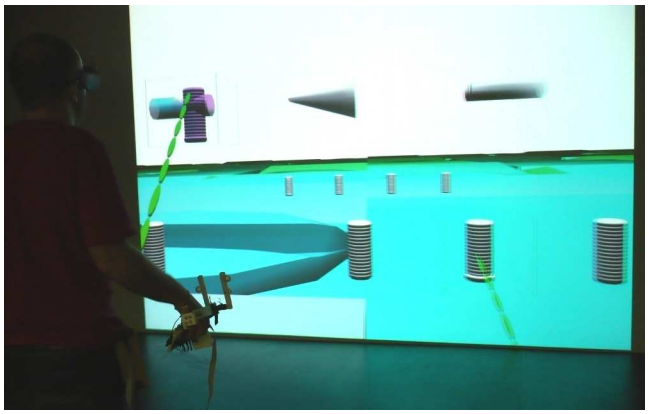


Figure 1: Immersive musical application using Piivert.

2 GENERAL APPROACH

We propose to split 3D musical interaction into several categories of musical gestures, i.e gestures that musicians do when playing an instrument, following Cadoz's [3] research. Cadoz defines selection gestures, modification gestures and excitation gestures.

Selection gestures are gestures by which musicians select instruments components, like the choice of the string on a violin. *Modification gestures* can be separated into structural modification gestures, which modify the structure of the instrument, and parametric modification gestures, which are also called modulation gestures. Modulation gestures modify the properties of an instrument, such as the action of the left hand on a violin or guitar. They can be discrete as with a guitar, or continuous as with a trombone. *Excitation gestures* directly and physically generate the sound, such as picking the string of the violin. They can be instantaneous, continuous or maintained.

We believe that separating usual 3D interaction into three gestures categories, i.e selection, excitation and modulation gestures,

*e-mail:berthaut@labri.fr
†e-mail:hachet@labri.fr
‡e-mail:myriam@labri.fr

may improve the interaction with immersive musical applications.

Selection gestures may be done graphically, using ray-casting technique [2]. Hardware music controllers, equipped with rotary knobs, also allow the selection of sound components, but they have a limited range, and they do not give much information on the selected components. On the contrary, graphical interfaces give access to an infinite number of graphical elements. Each of these elements can give meaningful visual clues on their associated sound processes and their parameters. Selection gestures also require less temporal accuracy than excitation gestures. They will not be disrupted by latency or tracking problems that may occur with graphical interaction.

Modulation gestures may also be done graphically, to benefit from the numerous tweakable visual parameters of graphical elements, such as shape, 3D position, texture and so on, to modify sound processes parameters. Moreover, one may easily change between continous and discrete scales graphically, to provide both discrete and continuous modulation gestures. When gaining expertise, these modulation gestures may be done more efficiently using hardware sensors with haptic feedback.

For excitation gestures, expert modulation gestures, and other controls that require accuracy and expressivity, graphical interaction may lead to inaccuracy problems and restrict musical interaction possibilities, for example by preventing combination with other musical gestures. High-efficiency tracking systems may partially solve the problem but they usually lack haptic feedback, which is difficult to setup in virtual reality centers in which movements of users should not be constrained. This feedback is however fundamental for music as explained in [9]. For excitation gestures, hardware sensors used in hardware musical controllers are better suited since they provide better sample rate and quantization, combined with passive haptic feedback. Using these sensors, musicians may be able to perform expert gestures, i.e gestures that are expressive and accurate.

There are many musical instruments and thus many existing musical gestures that can be used as these expert gestures. In a first approach to validate our concept, we chose to focus on percussion gestures. The first reason is that there is a large amount of them available. For example, in hand percussions, one can hit the drum skin at different points, scratch it or press it to modulate the pitch of the sound. The second reason is that these gestures are easy to obtain from hardware sensors [13], for example pressure sensors, with low latency and high quantization.

Based on these considerations, we developed Piivert, a new input device that combines graphical interaction and percussion gestures. This device is associated to a set of new interaction techniques, which may improve interaction with immersive instruments such as the example depicted in figure 1.

3 PIIVERT DEVICE

Our current prototype, as presented in figure 2 is strapped to the hand. It allows comfortable manipulation of the virtual ray using the wrist, while leaving the fingers free for percussive interaction.

The 6DOF tracking is achieved using A.R Tracking DTrack[1] system. Reflective targets are fixed on the devices and detected by two infrared cameras. The data is analysed in a dedicated computer and sent to the graphical application using the VRPN protocol [12]. The maximum frame rate of this system is 60 fps.

As explained in section 2, graphical interaction using tracking may lead to inaccuracy and thus restrict musical possibilities. In order to keep critical gestures independent of the display process,

[1]http://www.ar-tracking.de



Figure 2: Current prototype.

we use hardware sensors that give higher temporal accuracy. Force sensitive resistors (fsr) are used to get the percussion excitation gestures so that both pressure and hits can be detected [13]. Sensors are placed under the thumbs, index fingers, middle fingers and ring fingers of each hand. The little finger is not used because it appears to be less easily usable for percussion gestures. The acquisition is done with an Interface-Z[2] sensor board which transmits the values to a soundcard using Musical Instrument Digital Interface (MIDI) messages. The current prototype, as it can be seen in figure 2, is linked to a box that embeds the batteries and the sensor board. This box is hooked on the user's belt so that only one MIDI cable goes from the musician to the soundcard. The Interface-Z sensor board can be configured to send up to 1500 samples per second for all sensors. If the eight sensors are used simultaneously, this gives approximatively 187 samples per second, thus a latency of 5ms. Each data sample is 12 bits wide, i.e 4096 values, and it is obtained by mixing two 7 bits MIDI Control Change messages. The sensor board is connected to an external usb sound card, and MIDI messages are transmitted to the audio application using the ALSA MIDI driver and sequencer.

The complete setup of an example immersive musical application using our device can be seen in figure 3.

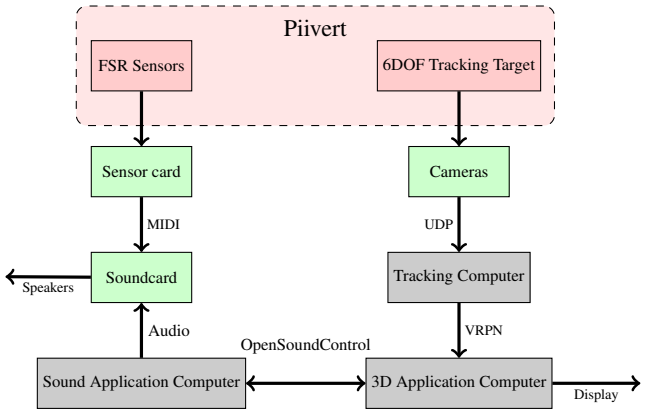


Figure 3: Immersive musical application setup.

4 PIIVERT INTERACTION TECHNIQUES

We developed a set of interaction techniques associated with Piivert. These techniques combine percussion gestures that we get from the fsr sensors and graphical interaction.

[2]http://www.interface-z.com/

4.1 Percussion gestures for excitation and control gestures

Percussion gestures are extracted from fsr sensors values. We first identify low-level gestures and then their combinations in higher-level gestures. In our current implementation, each gesture is sent to a callback function as a *piivert_event_** containing the type of the event, the finger(s) used and the value of the event.

4.1.1 Low-level percussion gestures

From the acquired data for each fsr sensor, i.e each finger, we can extract three low-level percussion gestures. The first one is the hit gesture, defined as *piivert_event_hit*. It is detected as a jump in the fsr sensor's values from 0 to at least 100 in less than 50ms. These threshold values were defined empirically after some testing. The value of the hit, i.e the velocity, is the first maximum following the jump. The second kind of gestures are pressure gestures. These are detected as any slow variation of fsr sensors' values. For example, they appear when sensors' values raise from 0 but do not reach the hit threshold fast enough. They are also detected as the non-zero values of a sensor following a hit gesture, typically when the user keeps pressing the sensors after having hit it. During a pressure gesture every value of the sensor is sent with a *piivert_event_pressure* event. Finally, a *piivert_event_release* is detected when the sensor's value gets back to 0, after a hit or pressure gesture. The value of the release gesture is the last non-zero value. We use a simple notation to describe these simple gestures, to make them more understandable. This results in a score for interaction gestures, which is depicted in figure 4. These low-level gestures may each be used for different excitation gestures. For example, hit gestures may play sound files associated to 3D audiovisual objects, loudness of the sounds being set by gestures' values. Pressure gestures may be used to control granular synthesis, which consists in reading short "grains" of the sound file in random order, with grains' positions set by gestures' value. Moreover, each finger can be associated to different sound synthesis parameters.

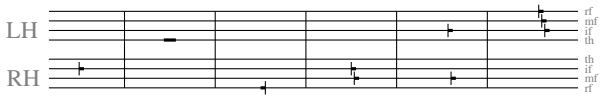


Figure 4: Low-level and high-level percussion gestures notation (RH: right hand, LH: left hand, th/if/mf/rf: fingers). From left to right : Hit, Pressure, Release, one hand Flam, two hands Flam, and three strikes Roll

4.1.2 High-level percussion gestures

High-level gestures can be detected as combinations of low-level gestures. These gestures allow us to trigger various actions such as changing the musical samples being used, or controlling musical sequences. The maximum time difference between two low-level gestures, so that they are not only understood as two low-level gestures, but also as part of a higher-level gesture, is 50 ms. In our current implementation, three high-level gestures are defined. The *piivert_event_flam* corresponds to a short sequence of two *piivert_event_hit*, which is known as a "flam". A short sequence of three *piivert_event_hit*, which is also known as a "three strikes roll", will be identified as a *piivert_event_roll3*. Finally, a *piivert_event_roll4* will be sent when a sequence of four *piivert_event_hit*, called a "four strikes roll", is detected. Examples of these high-level gestures are shown in figure 4.

Actions triggered by these high-level gestures integrate well with low-level gestures since they are also percussion gestures. They may thus be inserted in rythms, which may result in smoother and faster interaction after some practicing. One must also note that, in contrast to events triggered by simple buttons, high-level gestures have several usable parameters. First of all, the fingers used to make a gesture are identified so that one can make a distinction between a *piivert_event_flam* with the ring finger before the middle finger, and a *piivert_event_flam* in the other way. Each different fingers combination for each gesture may then be assigned to a different action. Bimanual gestures can thus be easily defined to get more meaningful actions, or to allow faster gestures sequences. These percussion gestures all have velocity, defined as the value of its last low-level gesture, and duration parameters, which may bring additional arguments for various actions, or add possibilities for musical interaction. An example of use for these parameters is that one may start an animation of some sound parameters with a flam, the velocity controlling the speed of the animation, and stop it using a flam in the other direction.

The velocity parameter of a high-level gesture is still 12 bits wide. The duration parameter of a gesture is between 10 and 50 ms for a *piivert_event_flam*, between 20 and 100 ms for a *piivert_event_roll3*, and between 30 and 150 ms for a *piivert_event_roll4*.

Beside the three high-level gestures defined above, one can also use combinations of low-level gestures. For example, simultaneous *piivert_event_pressure* and *piivert_event_hit* are similar to what can be done with many percussions, when musicians press the drum head to modify its tension and thus change the pitch of the sound while playing.

4.2 Graphical interaction for selection and modulation gestures

Selection gestures are performed graphically, using the ray-casting technique. Musicians manipulate two virtual rays, one in each hand, as bimanual gestures are common in musical instruments. The ray-casting technique is known to be very simple to use, and very efficient for fast selection of near objects in virtual environments [10]. By using this technique for selection gestures, one can directly point 3D graphical elements corresponding to sound processes, i.e audiovisual objects.

Because playing musical sequences may require to quickly select small objects distant from each other, one may need to perform very fast and large movements while keeping good temporal and spatial accuracy. Our implementation uses software aids to improve these selection gestures.

First we define larger bounding boxes for the audiovisual objects to be selected, as depicted on figure 5. Smaller movements are then needed to select the objects, because musicians do not have to point exactly on the objects. Another possible strategy is to divide the entire virtual environment according to the audiovisual objects. Hence the closest object to the virtual ray is automatically selected. A third method to minimize movements and improve efficiency of selection gestures is to snap the virtual ray to the closest object as proposed in [4].

With our current prototype, hit gestures may induce virtual ray jittering. This results in possible loss of the selected object. To overcome this ergonomic limitation, ray's movements are low-pass filtered when an audiovisual object is being played. This approach is similar to the anisomorphic ray-casting technique proposed in [1].

In addition to these techniques, we use another method to adapt ray-casting to musical interaction, the vibrating ray. When excitation gestures are performed to excite the audiovisual objects, vibrations are displayed as propagating on the virtual ray. This visual feedback improves the integration of musical gestures in the immersive environment. The vibrating ray can be seen in figure 5.

Playing very fast musical sequences using one virtual ray requires some training even when using the techniques described above. To facilitate musical interaction, we can take advantage of the fact that there is a fsr sensor for each finger on Piivert. We propose to split the virtual ray into two or three sub-rays, and to assign them to the fingers, so that they can be excited separately. Musicians can enter a selection mode, and select up to three objects with one ray. When re-entering any of the selected objects, the ray is then split accordingly, each sub-ray grabbing one object, as seen in figure 5. For example, if the ray is split in three, the index, middle and ring fingers may grab three selected audiovisual objects, that may respectively be associated to snare, kick and hi-hat sounds. Performing hit gestures with the sensors of those three fingers then allows to easily play drums rythms. This split ray technique allows us to select any group of two or three audiovisual objects of the environment, and to excite each of them easily, as one would do with a keyboard.

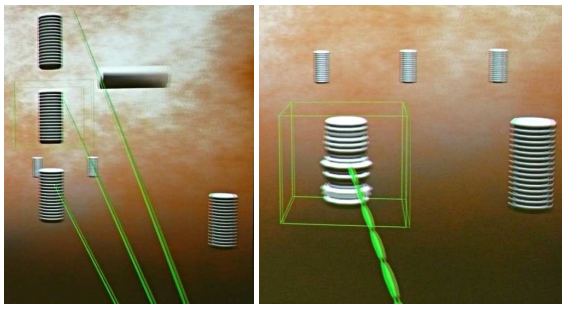


Figure 5: Split ray and vibrating ray techniques.

After one has selected an audiovisual object, he may grab it using a low-level pressure gesture with the thumb. Modulation gestures can then be defined as manipulation of the 3D objects, with different mappings. For example, position on the z-axis can be associated to the amplitude of the sound process, so that moving the object away reduces its volume. Position on the x and y axes may be respectively associated to sound processes' panning and pitch. One may also graphically modify other parameters of the audiovisual object, such as color hue, transparency, 3D shape. All these graphical parameters can be mapped to different musical parameters.

With Piivert it is possible to "excite" audiovisual objects with low-level gestures while performing these modulation gestures. This is similar to what one would do on a guitar by picking a string and sliding a finger on this string at the same time. Combined with the split ray technique described above, this allows us to trigger and modulate simultaneously up to three 3D audiovisual objects with each hand, as depicted on figure 5. This is almost impossible to do, or at least far less versatile with hardware musical instruments, as the selection and modulation of multiple objects is made possible by the graphical interaction.

5 Conclusion

In this paper, we presented a new input device, Piivert, and associated interaction techniques for immersive instruments. They rely on the separation of 3D interaction into musical gestures categories, i.e selection gestures, modulation gestures, and excitation gestures. This allows us to combine the possibilities of graphical interaction with the accuracy and expressivity of musical controls, here percussion gestures.

The work presented in this paper validates the foundations of a new approach for efficient musical interaction in immersive virtual environments. In the future, we want to push this concept more in depth. For example, we plan to improve the interaction techniques by first taking into account additional parameters, such as all finger velocities and velocity range in high-level events, and by allowing other high-level events. The device has also shown some ergonomic limitations which could be overcome using different materials and mechanisms. Numerous other directions are still to be explored.

Piivert has been developed for musical needs. However, we are convinced that such an approach may be valuable in other contexts, where efficiency and accuracy are required. This is the case for applications where time plays an important role, such as interactive simulations, visual performances and video games. Beyond efficiency and accuracy, we think that our approach opens new directions for interaction with 3D content. Indeed, the classical hardware buttons or software widgets that are currently used to control the system could be replaced by high-level gestures. The high-level parameters described in section 4.1.2 can add semantic links between the performed gestures and the corresponding graphical actions. We believe that such a percussion-based interaction have an interesting potential for future interactive applications.

References

[1] C. Andujar and F. Argelaguet. Virtual environments: Anisomorphic ray-casting manipulation for interacting with 2d guis. *Computer Graphics*, 31(1):15–25, 2007.

[2] D. A. Bowman and L. F. Hodges. An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. In *SI3D '97: Proceedings of the 1997 symposium on Interactive 3D graphics*, pages 35–ff., New York, NY, USA, 1997. ACM.

[3] C. Cadoz. *Musique, geste, technologie*. Éditions Parenthèses, 1999.

[4] G. de Haan, M. Koutek, and F. H. Post. Intenselect: Using dynamic object rating for assisting 3d object selection. In *IPT / EGVE 2005*, pages 201–209, Oct. 2005.

[5] S. de Laubier and V. Goudard. Meta-instrument 3: a look over 17 years of practice. In *NIME '06: Proceedings of the 2006 conference on New interfaces for musical expression*, pages 288–291, Paris, France, 2006. IRCAM — Centre Pompidou.

[6] S. Jordà. Interactive music systems for everyone: exploring visual feedback as a way for creating more intuitive, efficient and learnable instruments. In *Proceedings of the Stockholm Music Acoustics Conference (SMAC03)*, 2003.

[7] T. Mäki-Patola, J. Laitinen, A. Kanerva, and T. Takala. Experiments with virtual reality instruments. In *Proceedings of the 2005 International Conference on New Interfaces for Musical Expression (NIME05), Vancouver, BC, Canada*, 2005.

[8] A. G. Mulder. *Design of virtual three-dimensional instruments for sound control*. PhD thesis, 1998.

[9] M. S. O'Modhrain. *Playing by feel: incorporating haptic feedback into computer-based musical instruments*. PhD thesis, Stanford, CA, USA, 2001.

[10] I. Poupyrev, S. Weghorst, M. Billinghurst, and T. Ichikawa. *Egocentric object manipulation in virtual environments: Empirical evaluation of interaction techniques*. 1998.

[11] X. Rodet, F. Gosselin, P. Mobuchon, J.-P. Lambert, R. Cahen, T. Gaudy, and F. Guedy. Study of haptic and visual interaction for sound and music control in the phase project. In *Proceedings of the 2005 International Conference on New Interfaces for Musical Expression (NIME05), Vancouver, BC, Canada*, 2005.

[12] R. M. Taylor, II, T. C. Hudson, A. Seeger, H. Weber, J. Juliano, and A. T. Helser. Vrpn: a device-independent, network-transparent vr peripheral system. In *VRST '01: Proceedings of the ACM symposium on Virtual reality software and technology*, pages 55–61, New York, NY, USA, 2001. ACM.

[13] A. R. Tindale, A. Kapur, G. Tzanetakis, P. Driessen, and A. Schloss. A comparison of sensor strategies for capturing percussive gestures. In *NIME '05: Proceedings of the 2005 conference on New interfaces for musical expression*, pages 200–203, Singapore, 2005. National University of Singapore.

[14] M. Waisvisz. The hands. In *Proceedings International Computer Music Conference*, pages 313–318, 1985.