



**HAL**  
open science

# Construction, manipulation et visualisation de processus sonores dans des environnements virtuels immersifs pour la performance musicale

Florent Berthaut

## ► To cite this version:

Florent Berthaut. Construction, manipulation et visualisation de processus sonores dans des environnements virtuels immersifs pour la performance musicale. Informatique [cs]. Université Sciences et Technologies - Bordeaux I, 2010. Français. NNT : . tel-00573382

**HAL Id: tel-00573382**

**<https://theses.hal.science/tel-00573382>**

Submitted on 3 Mar 2011

**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.

# **Construction, manipulation et visualisation de processus sonores dans des environnements virtuels immersifs pour la performance musicale**

## **THÈSE**

présentée et soutenue publiquement le 7 octobre 2010

pour l'obtention du

**Doctorat de l'Université de Bordeaux**  
(spécialité informatique)

par

**Florent BERTHAUT**

### **Composition du jury :**

<i>Président :</i>	Pascal GUITTON	- Université de Bordeaux / INRIA
<i>Rapporteurs :</i>	Dominique BECHMANN	- Université de Strasbourg
	Marcelo WANDERLEY	- McGill University
<i>Examineur :</i>	Sergi JORDÀ	- Universitat Pompeu Fabra
<i>Directrice de thèse :</i>	Myriam DESAINTE-CATHERINE	- Université de Bordeaux
<i>Co-directeur de thèse :</i>	Martin HACHET	- INRIA



# Résumé

Le travail présenté dans cette thèse s'inscrit à la fois dans le domaine de l'informatique musicale et dans celui de la réalité virtuelle.

Nous présentons tout d'abord les avantages des environnements virtuels immersifs et de l'interaction graphique 3D pour les instruments multi-processus et en particulier pour le *live-looping hiérarchique*, un nouveau modèle d'instrument que nous définissons.

Nous proposons ensuite des solutions pour la visualisation et la manipulation des processus sonores dans ces environnements virtuels. Afin de représenter les processus sonores, nous introduisons les *widgets réactifs 3D* qui s'appuient sur des mappings audiovisuels que nous évaluons. Nous développons un nouveau dispositif d'interaction 3D adapté à l'interaction musicale, appelé *Piivert*, ainsi que des techniques associées, dont les *tunnels*, afin de manipuler les *widgets réactifs 3D*.

Finalement, nous mettons en œuvre le *live-looping hiérarchique* avec l'instrument immersif *Drile*, en utilisant les techniques développées précédemment, et nous étudions plusieurs scènes pour les performances musicales immersives.



# Abstract

The work presented in this thesis belongs to the computer music field as well as to the virtual reality field.

First, we describe the advantages of immersive virtual environments and of 3D graphical interaction for multiprocess instruments and particularly for the *hierarchical live-looping*, a new instrument model that we define.

We then suggest solutions to visualize and manipulate sound processes in these virtual environments. In order to display the sound processes, we introduce the *3D reactive widgets*, which rely on audiovisual mappings that we study. We develop a novel 3D interaction device suitable for musical interaction, called *Piivert*, together with associated interaction techniques, among which the *tunnels*, in order to manipulate the *3D reactive widgets*

Finally, we implement the *hierarchical live-looping* with an immersive instrument called *Drile*, using the previously developed techniques, and we study different stages for immersive musical performances.



# Table des matières

<b>1</b>	<b>État de l'art</b>	<b>15</b>
1.1	Interaction musicale . . . . .	15
1.1.1	Évolution . . . . .	15
1.1.2	Structure des instruments numériques . . . . .	16
1.1.3	Classification des instruments numériques . . . . .	21
1.2	Interaction graphique musicale . . . . .	24
1.2.1	Historique . . . . .	24
1.2.2	Spécificités . . . . .	25
1.2.3	Composants Graphiques . . . . .	26
1.2.4	Dispositifs pour l'interaction graphique musicale . . . . .	30
1.3	Réalité virtuelle . . . . .	33
1.3.1	Définition . . . . .	33
1.3.2	Historique . . . . .	33
1.3.3	Immersion . . . . .	33
1.3.4	Techniques et dispositifs . . . . .	35
<b>2</b>	<b>Instruments multi-processus</b>	<b>39</b>
2.1	Instrument multi-processus . . . . .	39
2.1.1	Définition . . . . .	39
2.1.2	Limitations des contrôleurs matériels . . . . .	42
2.1.3	Avantages de l'interaction graphique . . . . .	43
2.2	Live-looping hiérarchique . . . . .	44
2.2.1	Live-looping . . . . .	44
2.2.2	Live-looping Hiérarchique . . . . .	47
2.3	Environnements virtuels immersifs . . . . .	53
2.3.1	Environnements virtuels . . . . .	53
2.3.2	Instruments immersifs existants . . . . .	54
2.3.3	Problématiques . . . . .	56
<b>3</b>	<b>Représentation de processus sonores</b>	<b>59</b>
3.1	Widgets Réactifs 3D . . . . .	59
3.1.1	Principe . . . . .	59
3.1.2	Avantages pour les instruments multi-processus . . . . .	61
3.2	Combinaison de mappings audiovisuels . . . . .	63

3.2.1	Paramètres Sonores . . . . .	63
3.2.2	Paramètres Graphiques . . . . .	68
3.2.3	Travaux Existants . . . . .	71
3.2.4	Tests utilisateurs : Préférences, Performances et Combinaisons	73
3.2.5	Analyse et Discussion . . . . .	80
<b>4</b>	<b>Interaction</b>	<b>87</b>
4.1	Piivert . . . . .	87
4.1.1	Motivation : Limitations des dispositifs d'interaction 3D . . . .	87
4.1.2	Approche . . . . .	89
4.1.3	Dispositif . . . . .	91
4.1.4	Techniques d'interaction . . . . .	94
4.1.5	Évaluation de Piivert . . . . .	101
4.1.6	Perspectives . . . . .	105
4.2	Tunnels . . . . .	107
4.2.1	Motivation . . . . .	107
4.2.2	Principe . . . . .	107
4.2.3	Perspectives . . . . .	111
<b>5</b>	<b>Drile</b>	<b>113</b>
5.1	Drile . . . . .	113
5.1.1	Mise en œuvre . . . . .	113
5.1.2	Hiérarchie . . . . .	115
5.1.3	Pièces . . . . .	118
5.1.4	Apprentissage . . . . .	121
5.1.5	Collaboration . . . . .	121
5.1.6	Perspectives . . . . .	122
5.2	Performances musicales immersives . . . . .	124
5.2.1	Performances musicales immersives . . . . .	124
5.2.2	Scène Mixte . . . . .	126
5.2.3	Scène Virtuelle . . . . .	130
5.2.4	Perspectives . . . . .	131
	<b>Appendices</b>	<b>145</b>

## Remerciements

Je tiens tout d'abord à remercier Myriam Desainte-Catherine et Martin Hachet, qui m'ont encouragé à faire cette thèse et avec qui j'ai appris beaucoup pendant ces trois années ainsi que les membres du jury : Dominique Bechmann, Marcelo Wanderley, Sergi Jordà et Pascal Guitton.

Merci également :

Aux gens du bureau 327 : Luc, Raph et Antoine, puis Stan et Noémie.

À l'équipe Iparla (Sebastian, Benoit, Pascal, Nico, Fabrice, Mickael, Jérôme, Romain) pour les coups de main et les déménagements.

Au thème Son : Pierre, Thomas, Robert, Mathias, Sylvain, Joan.

Au Scrim et en particulier Joseph et Annick.

À Laurent pour les photos.

À ma famille, mes amis et surtout à Caro pour le soutien.



# Glossaire

**VRPN** : Virtual Reality Peripheral Network

**OSC** : Open Sound Control

**MIDI** : Musical Instrument Digital Interface

**DDL** : Degrés de libertés

**2D/3D** : Bidimensionnel/Tridimensionnel

**CAVE** : Cave Automatic Virtual Environment

**VBAP** : Vector Based Amplitude Panning

**WFS** : Wave Field Synthesis

**HMD** : Head Mounted Display



# Introduction

Au cours du XX<sup>ème</sup> siècle, les progrès de l'électronique puis de l'informatique ont permis la conception de nouveaux instruments de musique, ajoutant au contrôle de paramètres sonores tels que l'amplitude, la hauteur et le timbre, des possibilités de modification de paramètres musicaux de haut niveau sur des processus sonores. Le musicien électronique combine ainsi les rôles d'instrumentiste et de chef d'orchestre. Le contrôle d'instruments multi-processus, composés de plusieurs processus sonores s'exécutant simultanément, est grandement facilité par l'utilisation d'interfaces graphiques. Potentiellement, une infinité de processus représentés graphiquement peuvent être sélectionnés et manipulés.

Les interfaces graphiques 3D étendent ces possibilités. Elles offrent de nouveaux paramètres graphiques (formes 3D complexes, matériaux, éclairages, organisation spatiale) pour visualiser les paramètres sonores. Les nombreuses techniques d'interaction 3D permettant de sélectionner, déplacer et modifier des objets virtuels peuvent être adaptées à des fins musicales. La navigation dans des environnements 3D peut être utilisée comme métaphore efficace pour le parcours de partitions ou de structures musicales. Dans un contexte de réalité virtuelle, il est de plus possible d'immerger le musicien, et le public, dans un environnement musical. Cette immersion, en plus de l'aspect évidemment spectaculaire, peut améliorer la perception et la compréhension de l'instrument virtuel.

Néanmoins, hormis quelques instruments isolés, peu de travaux ont été menés sur l'utilisation des interfaces 3D immersives pour la performance musicale, plus spécifiquement dans le cas des instruments multi-processus. Tout au long de cette thèse nous avons cherché à tirer parti des possibilités de la réalité virtuelle tout en l'adaptant aux contraintes du contrôle musical. Nous nous sommes tout d'abord interrogés sur la représentation des processus sonores en 3D puis nous avons étudié comment interagir avec eux. Nous avons ensuite utilisé les solutions proposées pour mettre en œuvre un instrument de *live-looping hiérarchique*. Finalement, nous avons cherché à adapter notre instrument pour des performances publiques.

Ce mémoire est organisé en cinq chapitres.

Le chapitre 1 présente les domaines concernés. Il définit les spécificités de l'interaction musicale. Il décrit ensuite les caractéristiques de l'interaction graphique musicale et de la réalité virtuelle.

Le chapitre 2 expose l'intérêt de l'interaction graphique, et particulièrement des environnements immersifs, pour les instruments multi-processus. Il contient également la description d'un nouveau modèle d'instrument multi-processus, le *live-looping hiérarchique*.

Dans le chapitre 3, nous présentons les *widgets réactifs 3D*, qui permettent de représenter les processus sonores dans un environnement virtuel. Nous étudions ensuite les mappings audiovisuels utilisés pour ces widgets.

Le chapitre 4 décrit les techniques mises au point pour interagir avec des processus sonores dans un environnement virtuel. Nous présentons notamment les *tunnels* ainsi que le dispositif d'interaction *Pivert* et les techniques qui lui sont associées. Ce contrôleur permet de combiner les avantages de l'interaction graphique 3D et l'expressivité des contrôleurs musicaux.

Dans le chapitre 5 nous mettons en œuvre le *live-looping hiérarchique* avec l'instrument *Drile*, qui s'appuie sur les techniques proposées pour la représentation et la manipulation des processus sonores. Nous décrivons finalement plusieurs configurations de performances publiques permettant d'immerger correctement le public et les musiciens dans l'environnement musical, tout en préservant la compréhensibilité de l'interaction pour les spectateurs.

# Chapitre 1

## État de l'art

Ce chapitre décrit les domaines abordés dans ce mémoire, et en introduit les notions principales. La première section définit les caractéristiques de l'interaction musicale et des instruments numériques. La deuxième section se concentre sur les interaction graphique pour la musique. La troisième section présente le domaine de la réalité virtuelle et de l'interaction 3D.

### 1.1 Interaction musicale

Cette section présente l'interaction musicale. Elle décrit l'évolution des instruments acoustiques aux instruments numériques. Elle définit ensuite plus précisément ces instruments.

#### 1.1.1 Évolution

La grande majorité des instruments acoustiques impliquent une connexion directe entre le geste et la production sonore. L'énergie du geste est "convertie" en vibrations par l'instrument, vibrations qui se propagent dans l'air jusqu'aux tympans, vibrations que nous percevons comme des sons après traitements successifs de notre oreille interne. Cette conversion du geste en vibrations peut se faire par l'intermédiaire de cordes, de membranes ou de guides d'ondes. Les vibrations sont souvent amplifiées par l'utilisation d'un corps résonant.

Certains instruments ne présentent pas cette caractéristique de transmission directe. Ainsi, la cornemuse ou à un niveau moindre niveau les instruments à clavier sont caractérisés par une indirection dans ce flux d'énergie. Les gestes ne produisent alors plus directement les vibrations et peuvent être qualifiés de plus haut-niveau.

Avec l'évolution technologique, cette indirection s'accroît. Certains instruments tels que le Thérémine, mis au point par Léon Theremin en 1919, ou encore les Ondes Martenot, conçu par Maurice Martenot en 1918, en sont des exemples intéressants. La production sonore résulte de l'utilisation d'oscillateurs électroniques, le signal produit est amplifié et les vibrations sont produites par des membranes d'enceintes. A l'instar

des instruments acoustiques, ils ne permettent que le contrôle de paramètres de bas niveau, tels que la hauteur, l'amplitude et le timbre.

L'arrivée de la musique sur support apporte des contrôles de plus haut niveau, que se sont appropriés des compositeurs tels que Pierre Schaeffer. Le matériau musical peut être prélevé, modifié et réagencé temporellement.

Les instruments numériques achèvent la séparation du geste et du son. Ceci passe notamment par l'utilisation de protocoles musicaux tels que le MIDI [92] ou l'Open Sound Control [136]. Le musicien utilise des contrôleurs matériels complètement détachés des processus de synthèse. Les gestes, acquis grâce à différents capteurs comme pour les instruments électriques et électroniques, sont cette fois traduits en événements pouvant être des déclenchements de notes ou des modulations. Ces événements sont transmis, sous forme de messages du protocole musical, à des processus de synthèse sonore dans lesquels ils sont associés ("mappés") à des paramètres qu'ils modifient. Par exemple le Méta-Instrument [42] mis au point par Serge de Laubier permet de manipuler 54 paramètres simultanément, en utilisant plusieurs capteurs par doigt ainsi que des capteurs pour les mouvements des poignets et des bras. Le rapport entre le geste effectué et le son produit devient arbitraire : une simple pression peut déclencher la lecture d'une centaine de pistes musicales complexes.

Parallèlement, les processus sonores se diversifient. En plus des enregistrements dont le signal est maintenant échantillonné, d'autres types de synthèse sont créés ou du moins facilités. De même, plusieurs processus sonores peuvent être déclenchés et manipulés simultanément par un seul musicien, ce qui est évidemment beaucoup plus compliqué avec un instrument traditionnel, voire impossible suivant le nombre de processus. Ce principe est notamment très exploité dans les musiques électroniques.

Au cours de l'évolution vers les instruments numériques, une séparation s'est donc créée entre le geste du musicien et le son produit, mais cette séparation s'est accompagnée d'une complexification et d'une diversification de l'organisation des instruments.

### 1.1.2 Structure des instruments numériques

La figure 1.1 donne une vision schématique de l'organisation des instruments numériques, version simplifiée de l'organisation proposée par Jean-Michel Couturier [38].

Les gestes du musicien sont connectés à un module de contrôle, que l'on détaillera par la suite. Les paramètres de ce module sont à leur tour associés, "mappés", aux paramètres d'un module sonore, ce dernier pouvant être composé d'un ou de plusieurs processus sonores. Des informations sont également renvoyées au musicien, depuis le module de contrôle ainsi que depuis le module sonore.

Nous allons maintenant détailler les différentes parties de cette organisation, excepté le module contrôle qui diffère selon le type d'interaction. Il peut en effet être simplement un ensemble de capteurs regroupés sous forme de contrôleur matériel, tel les surfaces de contrôle MIDI, mais il peut également avoir un fonctionnement beaucoup plus complexe. Nous le décrirons par la suite dans le cas de l'interaction graphique.

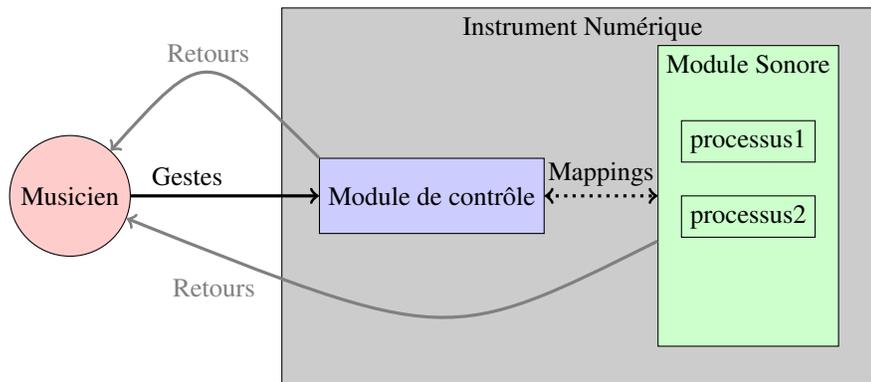


FIG. 1.1 – Organisation schématique d'un instrument numérique (ici contenant deux processus sonores).

### Geste instrumental

Le musicien interagit avec le module de contrôle par l'intermédiaire de gestes, souvent analogues aux gestes des instruments "traditionnels". Il existe plusieurs classifications du geste instrumental, applicables aux instruments numériques. Pour la suite de ce mémoire, et en particulier pour la section 4.1 nous utiliserons celle proposée par Claude Cadoz [28]. Ce dernier distingue 3 principaux types de gestes : les gestes de sélection, les gestes de modification, et les gestes d'excitation.

- Les gestes de sélection sont les gestes qui permettent au musicien d'accéder à différents composants de l'instrument, par exemple les différentes cordes d'une guitare. Ils peuvent être séquentiels comme dans le cas d'une suite de notes, ou simultanés pour un accord.
- Les gestes de modification sont des gestes qui changent les propriétés de l'instrument. On parle de gestes de modification structurelle lorsqu'ils modifient globalement l'instrument, et de gestes de modification paramétrique (ou gestes de modulation). Les gestes de modulation peuvent être discrets ou continus, suivant que le paramètre modifié est discrétisé comme les notes sur le manche d'une guitare, ou non.
- Les gestes d'excitation correspondent aux gestes qui créent physiquement le son, par exemple une frappe sur la peau d'une percussion. Ces gestes peuvent être instantanés dans le cas de gestes percussifs ou de pincements de cordes ou continus dans le cas de raclements. Dans ces deux cas, le son se prolonge après le geste. A contrario, les gestes d'excitation entretenus, par exemple pour les instruments à vent ou les instruments à archet, correspondent à des instruments pour lesquels le son n'est produit que pendant la durée du geste.

### Captation

Dans les instruments numériques, l'acquisition des gestes peut être réalisée de plusieurs façons comme le décrivent Wanderley et Depalle [128]. L'acquisition peut être directe, à l'aide de capteurs (de force, de déplacement linéaire ou angulaire, de vitesse, d'accélération), ou indirecte si le musicien utilise un instrument acoustique dont on analyse le son capturé à l'aide d'un microphone. Le choix des capteurs pour l'acquisition directe est déterminé par le geste à convertir, par le paramètre à contrôler et par l'organisation matérielle des capteurs. Ainsi, différents capteurs peuvent être utilisés pour les gestes de percussion comme décrit par Tindale et al. [115]. Les valeurs fournies par les capteurs sont ensuite numérisées. L'encodage de ces données et la latence sont deux caractéristiques importantes du système d'acquisition. La plupart des contrôleurs disponibles dans le commerce utilisent la norme MIDI avec une quantification des données sur 7 bits. Cela peut entraîner une discrétisation audible par des "sauts" pour certains paramètres tels que la fréquence de filtres. De plus pour certains gestes, par exemple les gestes d'excitation instantanés, la latence se doit d'être faible. En effet, au-delà d'un certain temps de latence, le musicien perçoit l'écart entre la frappe et le son résultant, et son jeu en est perturbé, particulièrement à des tempi rapides. Pour la musique ce seuil est généralement fixé à 10ms.

### Mappings

Les connexions entre le module de contrôle et le module sonore, peuvent être effectuées de différentes manières. Selon Hunt et Kirk [67], les mappings peuvent être de type *one-to-one*, *one-to-many* ou *many-to-one*. Le mapping *one-to-one* correspond à une association directe de deux paramètres. Par exemple, la position d'un encodeur linéaire peut être directement associée au volume d'une piste musicale. Le mapping *one-to-many* correspond à une association d'un paramètre en entrée à plusieurs paramètres en sortie. On retrouve ainsi ce qui se passe pour la plupart des instruments acoustiques. Par exemple, le fait de souffler plus fort dans un saxophone ne modifie pas seulement l'amplitude du son, mais également son timbre. Dans le mapping *many-to-one*, plusieurs paramètres d'entrée sont combinés dans un seul paramètre de sortie.

De plus, toujours selon Hunt et Kirk, les mappings peuvent être *explicites*, c'est à dire définis par des règles ou des fonctions, ou *implicites* car gérés par des "boîtes noires", par exemple des réseaux de neurones.

Différentes échelles (linéaires, logarithmiques, ...) et différentes opérations peuvent également être appliquées lors du mapping.

Finalement, un mapping peut se décomposer en plusieurs étapes comme le décrit Jean-Michel Couturier [38]. Une première étape de mapping peut amener des données des gestes à un espace de paramètres perceptif, plus simple, une deuxième étape connecte cette nouvelle série de paramètres à des paramètres sonores de haut niveau, qui sont à leur tour connectés à des paramètres sonores bas niveau. On trouvera également cette séparation en étapes dans le cas des interfaces graphiques musicales.

### Processus Sonores

Les processus sonores sont constitués de plusieurs types de composants : composant de synthèse audio, effets audionumériques, composants de contrôle, effets de contrôle. Ces derniers peuvent être agencés de différentes manières comme le montre la figure 1.2 Les composants de contrôle et les effets de contrôle produisent des flux de contrôle. Les composants de synthèse audio et les effets audionumériques produisent des flux audio.

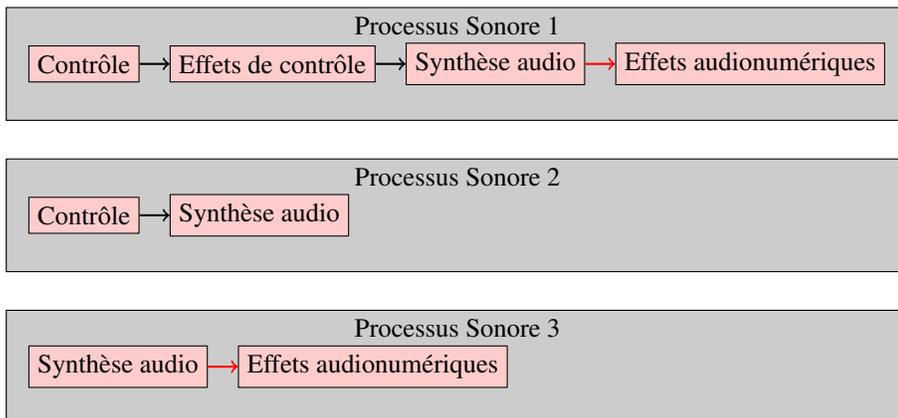


FIG. 1.2 – Quelques organisations possibles des composants des processus sonores. Dans le troisième cas, le composant de synthèse peut être une source audio externe, par exemple provenant d'un microphone.

**Synthèse Audio** Les composants de synthèse audio prennent en entrée uniquement des paramètres de contrôle et renvoient un signal sonore discrétisé. Le type de synthèse audio le plus simple est la lecture de sons pré-enregistrés. Les sons peuvent être découpés, répétés, réagencés temporellement ou encore lus à l'envers. Les paramètres sont ici relativement simples : déclenchement de la lecture, vitesse et sens de lecture, position de la tête de lecture.

La synthèse granulaire [106] se situe au croisement de la lecture de sons et de la synthèse à proprement parler. Cette technique consiste à jouer aléatoirement des petites parties (souvent entre 64 et 512 valeurs), appelés grains, d'un son enregistré. Cela donne souvent une impression de frottement sur le matériau sonore. Des résultats intéressants peuvent être obtenus en appliquant des effets audionumériques sur chacun des grains. Les paramètres de ce type de synthèse (taille/position/pitch... des grains) sont très nombreux.

La synthèse FM [34] utilise une fonction de modulation de la fréquence d'une sinusoïde. Cette fonction est simple mais permet néanmoins d'obtenir des résultats sonores très variés.

Les synthèses additive et soustractive, qui consistent respectivement à combiner des

sinusoïdes et à filtrer des bandes de fréquences à partir d'un signal, ont des paramètres beaucoup plus simples à manipuler, mais extrêmement nombreux.

La synthèse par modèles physiques [111] [29], qui s'appuie sur la reproduction des phénomènes physiques de production ou propagation du son, est intéressante car dans certains cas les paramètres de contrôles sont réduits à l'application de forces sur les structures virtuelles. L'analogie avec le monde réel simplifie son utilisation mais peut nécessiter plus d'expertise.

De même, la synthèse scannée [120] consiste à manipuler le contenu d'une table d'onde qui est scannée afin de produire le signal sonore.

Nous considérons également une source sonore externe, par exemple captée avec un microphone, comme un type de composant de synthèse audio.

**Effets audionumériques** Ces composants prennent en entrée un ou plusieurs signaux sonores et des paramètres de contrôle. Les paramètres peuvent être directement extraits du signal d'entrée dans le cas des effets adaptatifs développés par Verfaillie [119], ou contrôlés manuellement par le musicien. Ils dépendent évidemment de l'effet utilisé et seront donc plus ou moins compréhensibles. Par exemple, les paramètres liés à la perception humaine (hauteur, volume, brillance) sont relativement évidents à manipuler, comme nous l'expliquons dans la section 3.2.1.

**Contrôle** Les composants de contrôle correspondent le plus souvent à des séquences d'évènements de contrôle jouées soit une fois soit en boucle. Ces évènements peuvent être de plusieurs types :

- Déclenchements/Arrêts :  
Ces évènements permettent de contrôler la production du son par les composants de synthèse audio et également de déclencher les séquences des composants de contrôle. Le plus souvent ils sont associés à des notes musicales, hauteurs d'une échelle musicale (voir section 3.2.1), et permettent donc de jouer des mélodies. Dans la norme MIDI, les messages NOTE ON / NOTE OFF correspondent par exemple à des déclenchements/arrêts avec un numéro de note et une vélocité.
- Variations continues :  
Ces évènements permettent de manipuler les paramètres des différents composants. On les retrouve ainsi dans les pistes d'automation des séquenceurs multipistes afin de contrôler le volume ou d'autres paramètres des pistes audio. Dans la norme MIDI, ce sont les messages Control Change et AfterTouch.
- Variations discrètes :  
Ces évènements servent principalement à sélectionner des configurations avec un nombre restreint de valeurs. Dans la norme MIDI, ce sont les messages Program et Bank Change.

Ils vont être agencés sous forme de séquences MIDI, de pistes d'automation dans les séquenceurs multipistes. Les paramètres sont alors sensiblement les mêmes que pour la lecture de sons pré-enregistrés, à savoir déclenchement/arrêt, vitesse et sens de la lecture, position de la tête de lecture.

**Effets de contrôle** Tout comme le flux audio peut être modifié par des effets audio-numériques, le flux de contrôle peut également être modifié. Il est ainsi possible de filtrer/répéter/régencer/ralentir/modifier les événements de contrôle.

### Retours

Vertegaal et Ungvary [121] distinguent les retours primaires et les retours secondaires. Les retours primaires sont des retours avant traitement, par exemple le contact des doigts sur une touche de clavier, le son de cette touche, ou encore la vue du clavier. Les retours secondaires sont des retours après traitement, par exemple les sons produits par le module sonore, un retour visuel par interface graphique ou un périphérique à retour d'effort.

Les retours auditifs sont évidemment les retours les plus importants ici, le musicien ayant tout intérêt à pouvoir entendre ce qu'il joue ...

Les retours visuels primaires correspondent aux capteurs ou aux surfaces de contrôle que le musicien actionne, tandis que les retours secondaires sont des indicateurs lumineux de type LEDS, ou le plus souvent des interfaces graphiques. Les retours primaires et secondaires peuvent être superposés, par exemple par l'utilisation d'écrans tactiles ou d'environnements immersifs.

Finalement, les retours haptiques se déclinent en retours passifs et actifs [17] Les retours passifs correspondent en particulier au contact du musicien avec les contrôleurs matériels tandis que les retours kinesthésiques sont des mécanismes qui répondent aux actions de l'utilisateur. Ces retours actifs peuvent être mécaniques mais également vibrotactiles [86], ce qui est utilisé pour beaucoup de contrôleurs destinés aux jeux vidéos. Par exemple, le système développé par l'ACROE [48] est un ensemble modulaire de moteurs, sur un ou plusieurs axes, qui permet de "toucher" et manipuler des modèles physiques. Ce retour haptique est fondamental pour la musique [99]. S'il n'est pas toujours évident de mettre en œuvre un retour haptique actif, il est donc important d'au moins permettre un retour passif. Les retours pseudo-haptiques permettent néanmoins de le simuler par des techniques d'interaction graphique, qui donnent par exemple l'illusion de bosses ou de creux sur une surface en modifiant la vitesse d'un pointeur [78].

### 1.1.3 Classification des instruments numériques

La complexité et la diversité des nouveaux instruments engendre la nécessité d'une analyse et d'une classification. Les différentes classifications existantes s'appuient globalement sur les mêmes axes.

- Le premier axe est la complexité en entrée ou *degrés de contrôle*. Il représente la quantité de paramètres de l'instrument que l'utilisateur peut modifier. Franco et al. [49] insistent sur la nécessité d'équilibrer ces degrés de contrôle par rapport aux degrés de liberté des dispositifs d'interaction utilisés.
- Le deuxième axe est la liberté laissée par l'instrument au musicien, que Sergi Jordà nomme *diversity control*. Par exemple, un instrument doit donner la possibilité de mal jouer, et de jouer plusieurs fois la même pièce mais de manière différente. Cet axe rejoint la classification proposée par Pressing [101] prenant

en compte le rôle du son dans l'application. Il permet en particulier de séparer les applications de type installation interactive des vrais instruments. Pressing distingue ainsi trois types d'utilisation du son : artistique, environnementale et informationnelle.

- Le troisième axe est la complexité en sortie, c'est à dire le résultat musical. Sur un instrument composé de plusieurs processus sonores cela peut représenter le nombre processus sonores simultanés, dans les autres cas cela peut être simplement la diversité de sons que l'on peut obtenir avec l'instrument (hauteurs, amplitudes, timbres).
- Le quatrième axe est la courbe d'apprentissage. Ainsi, une utilisation virtuose, c'est à dire exploitant toutes les possibilités de l'instrument [44], doit être possible après un temps de pratique raisonnable, mais l'instrument doit rester accessible aux utilisateurs novices (leur permettre d'avoir un impact, même très simple, sur le son) [133] [49].
- Finalement, un axe important concerne les retours disponibles, qu'ils soient haptiques, visuels, auditifs. En effet suivant la quantité et la qualité de ces retours, l'instrument a plus ou moins de présence. Cela influence la perception qu'a le musicien de l'instrument, et les informations qu'il peut utiliser pour améliorer son jeu.

Donnons quelques exemples d'analyse/classification.

Sergi Jordà [72] définit une mesure de l'efficacité d'un instrument en fonction de la complexité en entrée et de la complexité en sortie. Il prend l'exemple du piano, complexe à jouer mais riche, par opposition au kazoo, simple mais donnant peu de possibilités musicales. Avec les instruments numériques, l'équation est un peu plus complexe. Le geste étant dissocié du processus musical, il est possible d'avoir un résultat très riche avec très peu de gestes. On peut ainsi imaginer déclencher la lecture d'une centaine de séquences musicales par la pression sur un seul bouton. Cet instrument est alors beaucoup plus efficace que tous les instruments existants, la complexité de l'interaction est quasiment nulle, et la complexité du résultat musical est potentiellement infinie. Sergi Jordà ajoute donc le paramètre de *diversity control* qui pénalise les instruments laissant peu de libertés aux musiciens. L'équation finalement obtenue est la suivante :

$$Efficiency = \frac{MusicalOutputComplexity \times DiversityControl}{ControlInputComplexity}$$

Wanderley et al. [127] proposent un schéma à 2 axes, l'un correspondant à l'expertise requise par l'instrument, l'autre correspondant au type d'interaction requis, des gestes à la manipulation, que l'on pourrait rapprocher de la notion d'influence du musicien. Ils distinguent ainsi à une extrémité les installations sonores, et de l'autre les instruments de musique qui se chevauchent avec les manipulations de réalité virtuelle.

A partir de notions similaires, Birnbaum et al. [14] utilisent des *dimension spaces*. Ils donnent un exemple de *dimension space* à 7 axes, que l'on peut voir sur la figure 1.3. Cette classification permet en particulier de rapidement visualiser l'orientation d'une interface musicale. En effet, un instrument très expressif tel que The Hands de Michel Waisvisz [125], qui se positionne plutôt vers le coin haut droit du schéma comme on

le voit sur la figure 1.4, se démarque clairement d'une installation (moins) interactive qui sera plutôt vers le coin bas gauche. On retrouve dans cette classification les axes *role of sound*, *musical control* (qui correspond à la liberté laissée par l'instrument au musicien car il donne le niveau de détails auquel le musicien a accès), *required expertise* (qui s'apparente à la courbe d'apprentissage), *degrees of freedom(input)*, *feedback modalities(output)*. S'ajoutent deux axes : *distribution in space* donne l'échelle de l'instrument(local petit/local grand/distant), *inter-actor* définit le nombre d'utilisateurs simultanés de l'instrument (ce nombre sera souvent faible pour les instruments).

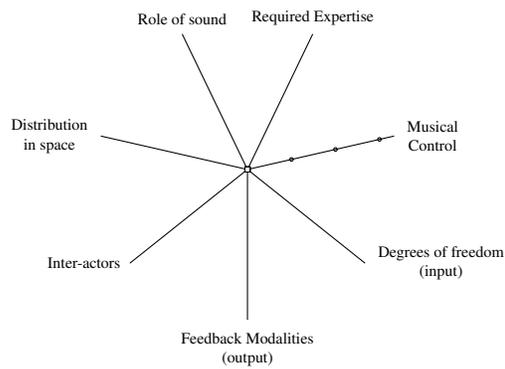


FIG. 1.3 – Dimension Space Axes

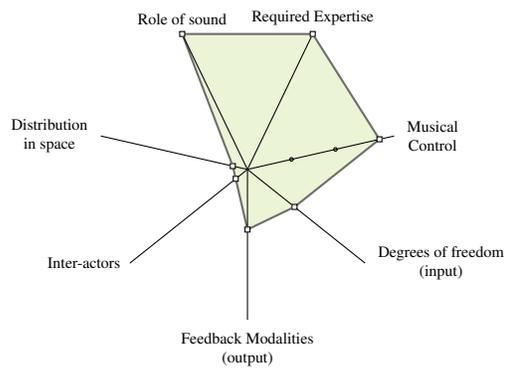


FIG. 1.4 – Dimension Space pour The Hands de M. Waisvisz [125]

## 1.2 Interaction graphique musicale

Les instruments numériques s'appuient de plus en plus sur des interfaces graphiques. Leur structure est alors celle présentée sur la figure 1.5. Le module de contrôle est constitué d'un dispositif d'interaction et de l'interface graphique.

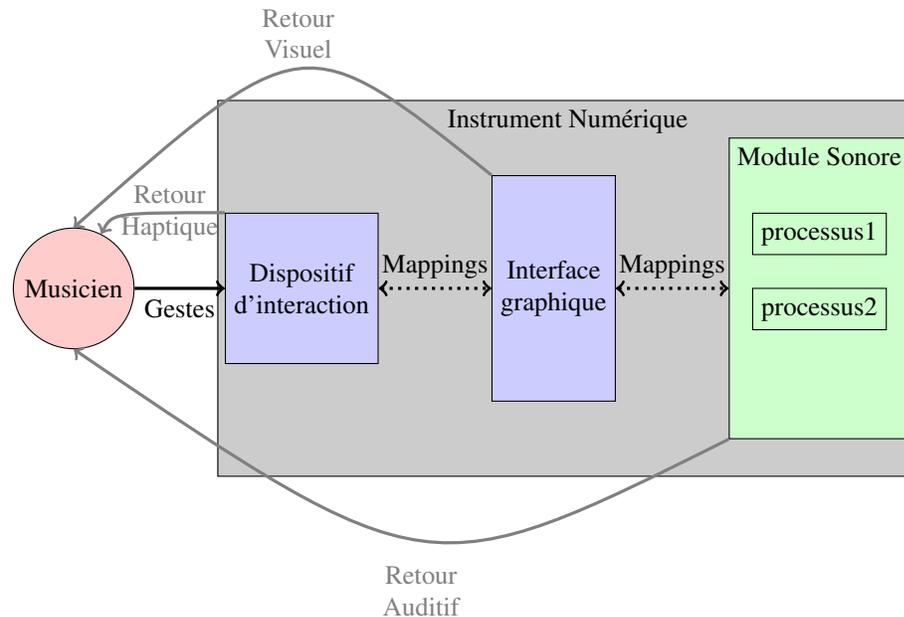


FIG. 1.5 – Organisation schématique d'un instrument numérique utilisant une interface graphique. Le schéma montre également les principaux retours Visuels, Auditifs et Haptiques

Dans cette section, nous présentons les origines de l'interaction graphique, puis les spécificités de l'interaction graphique musicale. Nous décrivons ensuite succinctement les principaux composants graphiques ainsi que les dispositifs et techniques d'interaction utilisés pour la musique.

### 1.2.1 Historique

L'interaction graphique débute en partie avec Ivan Edward Sutherland qui met au point le logiciel SketchPad [113] dans les années 60. Ce logiciel fait avancer l'interaction homme-machine de la communication textuelle à la communication graphique. Des métaphores graphiques pour la bureautique, toujours utilisées, apparaissent ensuite. En particulier le système NLS de Douglas Engelbart inaugure l'utilisation de fenêtres graphiques et du curseur graphique manipulé par une souris. Vient en suivant le PDS-1 premier ordinateur avec une interface graphique. Les interfaces WIMP (windows, icons, menus, pointers), dont on peut voir deux exemples sur la figure 1.6, sont

finalisées au Xerox Parc dans la fin des années 70, avec notamment le système Xerox Star. Elles sont destinées principalement à un usage bureautique : manipulation de documents représentés par des icônes à l'aide d'un curseur, modification de ces documents par l'intermédiaire de fenêtres graphiques et de fonctions organisées sous forme de menus. Le succès de ces interfaces tient notamment aux métaphores utilisées, qui s'adaptent naturellement à un grand nombre de logiciels.

Depuis cette époque, les interfaces grand public ont peu évolué, notamment du point de vue des métaphores, et ce sont les applications plus spécialisées (en particulier artistiques) qui ont vu le plus d'innovations.



FIG. 1.6 – Peu d'évolution entre les interfaces WIMP originelles (écran du Xerox Star à gauche) et actuelles (environnement de bureau XFCE à droite).

## 1.2.2 Spécificités

Les interfaces graphiques pour la création et la performance musicale se rapprochent plus des principes de la manipulation directe énoncés par Shneiderman [109] que des interfaces WIMP. Ces principes correspondent à l'avènement interfaces graphiques et s'opposent ainsi aux interfaces textuelles. Ils rejoignent le modèle de l'interaction instrumentale proposé par Beaudoin-Lafon [7] par la notion d'objets d'intérêts (objets que l'utilisateur veut modifier) qui doivent être tout le temps visibles et accessibles. Les opérations doivent également être effectuées directement sur les objets d'intérêts et non par le parcours de menus graphiquement éloignés.

Les analyses des spécificités et règles applicables aux interfaces graphiques musicales (ou systèmes audiovisuels) effectuées notamment par Vinet [122], Levin [80], Hunt et Kirk [67] ou encore Franco et al. [49] sont relativement similaires et peuvent être regroupées par problématiques. Certaines de ces spécificités rejoignent naturellement les dimensions énoncées pour les instruments numériques.

### **Temporalité**

La première spécificité est la notion de temporalité. Contrairement aux images, qui sont perçus en une fois, l'information sonore s'étale dans le temps. Comme le note Vinet, les interfaces musicales doivent donc prendre compte ce paradoxe d'instantanéité/séquentialité. A l'inverse, Hunt note que le contrôle n'a pas d'ordre fixé, il ne correspond pas à un enchaînement d'actions, et que le but de ces interfaces est plus un contrôle global qu'un transfert d'informations.

### **Contrôle**

Les interfaces musicales se caractérisent selon Hunt par un contrôle complet de l'utilisateur, qui initie toutes les actions, le système se contentant de répondre, de manière instantanée. Vinet parle également de staticité/dynamicité, les éléments graphiques sont figés sauf s'il y a manipulation. Les contrôles sont plutôt continus, et non un ensemble de choix discrets.

### **Malléabilité**

Les analyses se rejoignent également sur la notion de malléabilité, de flexibilité des interfaces graphiques musicales. Vinet décrit ainsi les principes de figuration/abstraction, les processus sonores peuvent être représentés efficacement de différentes manières. Levin insiste sur la nécessité pour l'utilisateur de pouvoir définir sa grammaire graphique, ses représentations. Selon Franco, ces interfaces doivent également offrir la possibilité de modifier les mappings audiovisuels, de créer et d'agencer les contenus sonores et visuels, et de disposer de modificateurs.

### **Apprentissage**

Par rapport aux interfaces graphiques génériques, pour lesquelles les travaux de recherche s'orientent vers des techniques d'interaction immédiatement efficaces, ou "naturelles" [25], les interfaces musicales, comme les instruments, nécessitent souvent un apprentissage. Hunt le justifie par le nombre élevé de paramètres à manipuler et par la présence de mappings complexes. La conception d'interfaces musicales rejoint la problématique des courbes d'apprentissage exposée dans la section 1.1 pour les instruments numériques.

## **1.2.3 Composants Graphiques**

Golan Levin [80] définit une classification pour les composants graphiques musicaux. Il distingue ainsi des composants de commande, de visualisation et d'organisation temporelle et structurelle. Les composants décrits se retrouvent aussi bien dans les interfaces 2D que 3D.

### Objets de commande virtuels

Les objets de commande virtuels sont utilisés pour la manipulation des paramètres des processus sonores. Ils imitent le plus souvent des composants matériels, tels que des potentiomètre linéaires ou angulaires et des boutons. La figure 1.7 présente un panneau de commande du logiciel LMMS, les différents potentiomètres permettent de modifier des paramètres de synthèse. Les potentiomètres permettent de manipuler un ou deux paramètres dans le cas des interfaces 2D. Au-delà de deux paramètres, il est nécessaire de passer par une interpolation entre *presets* de paramètres. C'est ce qu'utilisent pas exemple les logiciels GRM Tools<sup>1</sup>.



FIG. 1.7 – Panneau de commande d'un instrument du logiciel LMMS

### Objets de visualisation

Les objets de visualisation permettent d'afficher des paramètres physiques ou perceptifs du signal sonore, tels que le spectre, le volume, la spatialisation. Ils sont principalement utilisés pour contrôler l'état du signal lors de modifications de paramètres sonores, afin notamment d'éviter le clipping (valeurs du signal supérieures à la valeur maximale, entraînant une saturation), ou afin d'équilibrer les fréquences.

### Représentations temporelles

Ces composants permettent de visualiser et de modifier l'organisation temporelle de contenu musical. Ce contenu est le plus souvent un ensemble d'enregistrements sonores ou d'évènements de contrôle. Par exemple, les séquenceurs permettent souvent de définir des partitions sous forme de piano-roll, avec les notes en ordonnée et les

<sup>1</sup><http://www.grmtools.org/>

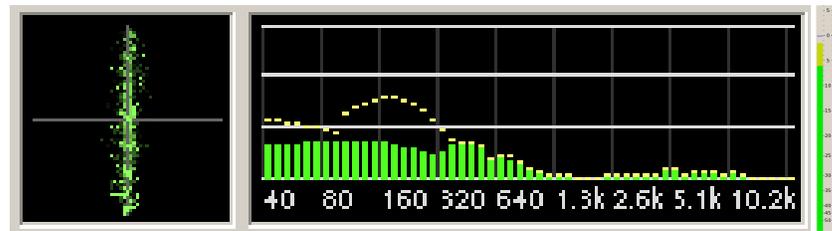


FIG. 1.8 – Composants de visualisation de la spatialisation et du spectre du logiciel Rezound et Composant de visualisation du volume (vu-mètre) du logiciel Jamin

temps en abscisse. L'organisation peut s'effectuer sur plusieurs niveaux, avec l'agencement de notes qui forment des motifs, puis l'agencement de ces motifs pour constituer des morceaux. Le plus souvent le temps est représenté par l'axe des abscisses, comme présenté sur la figure 1.9, (ou sur l'axe des ordonnées pour les logiciels de type tracker tels que Renoise<sup>2</sup>) mais il existe également des représentations circulaires adaptées aux éléments joués en boucle, et des représentations s'appuyant sur des trajectoires plus complexes comme le système UPIC proposé par le compositeur Iannis Xenakis, ou sa version plus récente appelée Iannix.

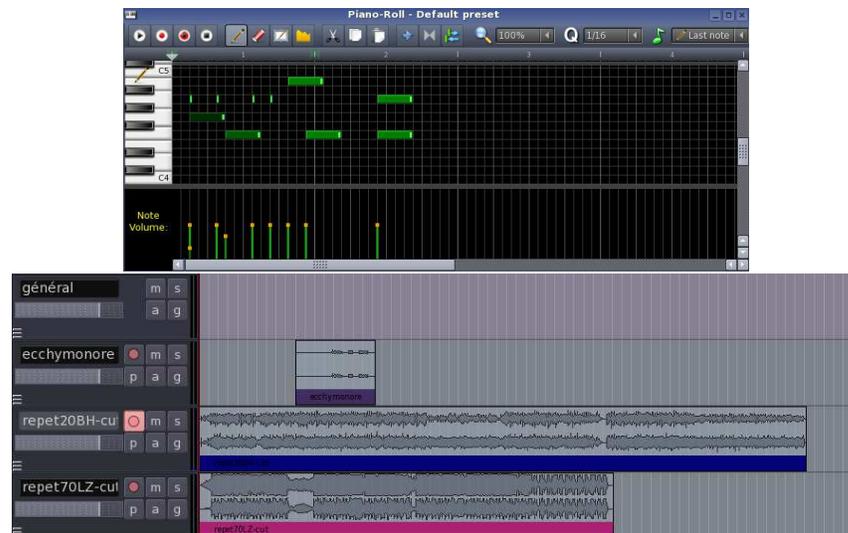


FIG. 1.9 – Piano-roll de LMMS et pistes du logiciel Ardour

<sup>2</sup><http://www.renoise.com/>

## Diagrammes

Les diagrammes permettent de visualiser et de modifier l'organisation logique des processus sonores. Ils sont constitués d'éléments représentant des composants logiciels (processus de synthèse, effets, autres composants graphiques) et d'éléments représentant les connexions entre ces composants. Par exemple, le logiciel Pure-Data, illustré par la figure 1.10 est constitué d'objets de traitement et d'objets graphiques, connectés entre eux par des flux de contrôle et des flux de signal.

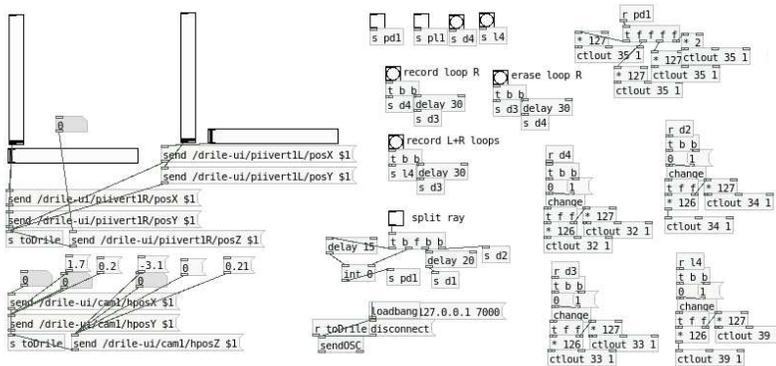


FIG. 1.10 – Patch du logiciel Pure-Data

## Widgets Réactifs

Les widgets réactifs, définis par Levin [80], sont des composants graphiques regroupant plusieurs propriétés des autres composants présentés. Ils ont un comportement complexe et réagissent de manière évoluée aux interactions car ils permettent à la fois de contrôler et de visualiser les processus sonores. Ils peuvent également interagir entre eux. Un exemple d'utilisation des widgets réactifs se trouve dans le logiciel FMOL [71], présenté sur la figure 1.11. Ce logiciel permet de contrôler des processus de synthèse en "grattant" des cordes virtuelles qui représente simultanément la forme d'onde du signal produit.

Les applications développées par la société Ixi Software [83] utilisent également les widgets réactifs.

## Dessin interactif

Les composants de dessin interactif développés par Levin [80], dont on peut voir un exemple sur la figure 1.12, par Franco et al. [49] ou encore par Couturier et Roy [39] diffèrent des autres composants graphiques de contrôle. En effet, ils ne sont pas visuellement figés et leur manipulation consiste en la création de contenu audiovisuel plutôt qu'en la modification d'éléments prédéfinis.

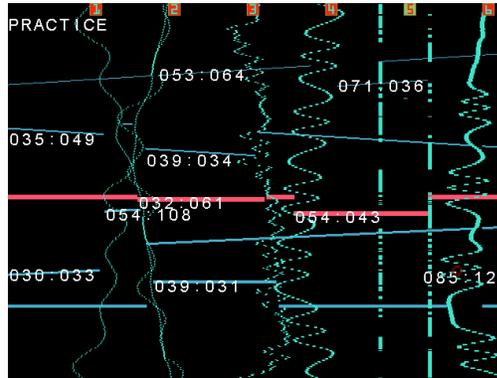


FIG. 1.11 – Logiciel FMOL de Sergi Jordà.

### 1.2.4 Dispositifs pour l'interaction graphique musicale

Les interfaces graphiques ont beaucoup été utilisées pour l'écriture (composition) avec les composants de représentation temporelle, mais elles le sont de plus en plus pour la performance, notamment grâce aux nouveaux types d'interaction graphique (tangible, 3D immersif ...).

#### Indirects 2D

Le dispositif le plus commun pour interagir avec les interfaces graphiques musicales reste la souris. L'interaction est alors dite indirecte, les retours visuels primaires et secondaires n'étant pas superposés. La souris permet de manipuler tous les composants graphiques, avec une bonne précision spatiale (positionnement au pixel), mais une précision temporelle plus réduite. La modification d'un composant graphique à la souris comporte de plus une phase d'activation, c'est à dire de pointage du composant [38], avant d'effectuer la manipulation. Plusieurs instruments ont été spécialement conçus pour une utilisation à la souris, par exemple MidiGrid [68].

#### Contrôleurs matériels

Très fréquemment, les interfaces graphiques sont utilisées conjointement à des contrôleurs matériels. Les paramètres des composants graphiques sont connectés aux capteurs de ces contrôleurs par différents mappings, comme nous le décrivons dans la sous-section 1.1.2. L'interface sert alors essentiellement à la visualisation des paramètres des processus (sélection de presets, états des processus), tandis que la plupart des manipulations sont effectuées grâce aux contrôleurs.

On distingue généralement trois catégories de contrôleurs. Les contrôleurs imitatifs ont l'apparence des instruments traditionnels mais ils ne produisent aucun son, ils captent les gestes effectués et envoient des messages de contrôle musicaux. On trouve

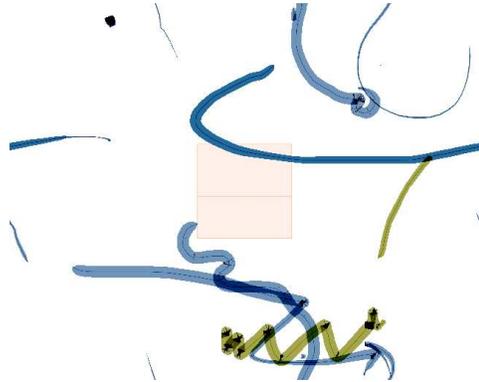


FIG. 1.12 – Logiciel Yellow Tail de Golan Levin. Les gestes (tracés) effectués sont rejoués en boucle et une tête de lecture virtuelle permet de les traduire en paramètres de synthèse audio additive.

ainsi des claviers, des instruments à vent, des batteries équipés de capteurs. Les contrôleurs analogues sont des instruments acoustiques auxquels sont ajoutés des capteurs. On parle également d'instruments augmentés. Les contrôleurs alternatifs sont des ensembles de capteurs dont la structure se différencie des instruments traditionnels. En dehors des surfaces de contrôle qui reprennent l'organisation des tables de mixage, il existe aussi des contrôleurs plus atypiques.

Dans la plupart des cas, l'affichage de l'interface est séparée du contrôleur, généralement sur un écran d'ordinateur à proximité du musicien. Dans d'autres cas, l'interface est projetée sur le contrôleur, ce qui permet de réduire l'indirection. Ainsi le Jam-O-Drum [15] permet à plusieurs utilisateurs d'interagir simultanément avec un instrument projeté sur une table circulaire à l'aide de capteurs de vibrations (pads électronique) et de potentiomètres (platine vinyl).

### Interfaces Tactiles et Multipoints

Les interfaces tactiles permettent de manipuler les composants graphiques de manière directe. Les retours visuels primaires et secondaires sont fusionnés et l'utilisateur "touche" les composants graphiques. Ceci peut cependant entraîner des problèmes d'occultation et de précision, notamment lorsque les manipulations s'effectuent avec les doigts, si les interfaces ne sont pas adaptées. En effet, l'utilisateur risque de masquer des composants graphiques, et la précision des paramètres graphiques est réduite par la taille des doigts.

En plus des interfaces tactiles simples, qui ne permettent qu'un seul contact (un seul doigt ou stylet) avec l'interface à la fois et qui sont couramment utilisées pour le dessin, il existe des interfaces tactiles multipoints. Celles-ci permettent de tirer partie

des interaction bimanuelles, et de concevoir de nouvelles techniques de manipulation d'éléments graphiques.

### **Interfaces Tangibles**

Les interfaces tangibles ont pour but de rétablir un retour haptique passif dans l'interaction graphique, en s'appuyant sur la manipulation d'objets réels superposés à des composants graphiques. Introduites par Fitzmaurice et al. [47], elles se sont beaucoup développées dans le domaine de l'interaction musicale, notamment car elles favorisent l'interaction collaborative et qu'elles se révèlent faciles à comprendre et à prendre en main. L'exemple le plus célèbre est certainement la Reactable [73], qui permet de construire des patchs dynamiques en agençant des objets associés à différents processus sonores.

### **Réalité virtuelle, Interaction 3D**

Les spécificités des environnements virtuels immersifs et des interfaces 3D sont détaillées dans la section 1.3. Les principaux instruments immersifs existants sont présentés dans la section 2.3.

## 1.3 Réalité virtuelle

### 1.3.1 Définition

Le terme "virtual reality" est introduit dans les années 80 par Jaron Lanier. Plusieurs définitions de la réalité virtuelle existent, les chercheurs de l'association française de réalité virtuelle (AFRV) s'accordent néanmoins sur les définitions suivantes :

*La finalité de la réalité virtuelle est de permettre à une personne (ou à plusieurs) une activité sensori-motrice et cognitive dans un monde artificiel, créé numériquement, qui peut être imaginaire, symbolique ou une simulation de certains aspects du monde réel. [3]*

*La réalité virtuelle est un domaine scientifique et technique exploitant l'informatique et des interfaces comportementales en vue de simuler dans un monde virtuel le comportement d'entités 3D, qui sont en interaction en temps réel entre elles et avec un ou des utilisateurs en immersion pseudo-naturelle par l'intermédiaire de canaux sensori-moteurs. [51]*

L'immersion est plus précisément définie par Slater [110] comme le *niveau objectif de fidélité perceptive*, c'est à dire les moyens mis en oeuvre pour "duper" les sens de l'utilisateur. La perception de cette immersion, c'est à dire le niveau subjectif de fidélité perceptive, correspond alors à la notion de présence.

### 1.3.2 Historique

La réalité virtuelle trouve évidemment ses origines conceptuelles dans les œuvres de science-fiction (Philip K. Dick, William Gibson). Elle ne sera rendu techniquement possible qu'avec les progrès en matière de synthèse d'image et de dispositifs d'affichages et d'interaction.

Parmi les travaux précurseurs, on peut citer Sutherland et son casque de visualisation Ultimate Display, représenté sur la figure 1.13, développé en 1968. Mais ce sont surtout les travaux menés de McGreevy à partir de 1984 au sein du programme VIVED (Virtual Visual Environment Display) qui lancent les recherches sur la réalité virtuelle.

Parallèlement aux dispositifs d'affichage immersifs se développent les premiers dispositifs d'interaction destinés aux environnements virtuels, notamment avec les travaux de Krueger et al. [76] et de Brooks et al. [27].

### 1.3.3 Immersion

L'immersion, comme définie dans la section 1.3.1, peut être séparée en immersion sensorielle et immersion cognitive. L'immersion cognitive résulte des possibilités d'interactions avec l'environnement virtuel. L'immersion sensorielle est principalement réalisée à trois niveaux : visuel, haptique et auditif.



FIG. 1.13 – Système Ultimate Display mis au point par Sutherland.

### Visuelle

L'immersion visuelle s'appuie tout d'abord sur des techniques de rendu permettant de restituer l'impression de profondeur dans l'environnement virtuel. Sont ainsi utilisés des masquages d'objets, des ombres projetées ou encore des modifications de l'éclairage pour donner des effets de relief aux textures (bump mapping, parallax mapping).

Mais la technique principale consiste à afficher un point de vue différent de l'environnement virtuel pour chaque œil de l'observateur, la plupart du temps à l'aide de lunettes, afin que le cerveau reconstitue le relief à partir des deux images. Les systèmes de rendu stéréoscopique peuvent être passifs (vues superposées et filtres de couleurs ou polarisation), actifs (synchronisation de l'affichage d'une vue et de l'ouverture d'un cache pour l'œil correspondant), ou encore autostéréoscopiques auquel cas les lunettes ne sont pas nécessaires mais la position de l'utilisateur par rapport à l'écran est contrainte.

En plus de la stéréoscopie, les systèmes de réalité virtuelle s'appuient souvent sur le suivi des mouvements de la tête (head-tracking) afin d'adapter le rendu au point de vue de l'utilisateur, ce qui lui permet par exemple de "tourner" (du moins partiellement) autour des objets 3D. Cette technique est habituellement limitée à une seule personne par affichage, la vue se modifiant suivant la position d'un utilisateur. Il est néanmoins possible de mettre en œuvre le suivi de mouvements de la tête pour plusieurs utilisateurs sur un seul affichage, en augmentant la fréquence d'affichage et en utilisant le même principe que la stéréoscopie active, comme le montrent les travaux de Fröhlich et al. [50]. Les différents systèmes de suivi, également utilisées pour l'interaction, sont décrites dans la sous-section 1.3.4.

### Haptique

L'immersion haptique consiste à ajouter un retour haptique (voir la section 1.1.2) le plus souvent aux dispositifs d'interaction. Celui-ci peut être actif par exemple dans

le cas de dispositifs à retour d'effort [19] [87] ou vibrotactiles, ou bien passif pour les autres dispositifs d'interaction (capteurs de pression, potentiomètres, ...).

### **Auditive**

L'immersion auditive consiste principalement à spatialiser, c'est à dire positionner dans l'espace, des sources sonores, le plus souvent associées à des objets de l'environnement virtuel. Cette spatialisation peut se faire de manière binaurale en utilisant un casque et les propriétés de la perception auditive (voir section 3.2.1), de manière transaurale avec deux haut-parleurs, ou avec un nombre supérieur d'enceintes grâce à la technique VBAP [103] ou à d'autres techniques. Ces méthodes sont cependant surtout efficaces pour un nombre restreint d'utilisateurs (voire pour un seul) à une position précise, contrairement à la synthèse de front d'onde (WFS) [8] qui restitue les ondes qu'émettraient des sources sonores virtuelles et pour laquelle la localisation n'est donc pas dépendante de la position d'écoute. Les effets de réverbération et de matériaux sont également souvent ajoutés afin de parfaire l'immersion.

## **1.3.4 Techniques et dispositifs**

### **Visualisation**

Les principaux dispositifs d'affichage sont représentés sur la figure 1.14. En plus de ces dispositifs, on trouve des applications de réalité virtuelle destinées à une utilisation de bureau, c'est à dire avec un écran d'ordinateur standard.

Les trois premiers dispositifs s'appuient habituellement sur des vidéo-projecteurs. La projection peut se faire par l'avant sur un écran opaque ou par l'arrière sur un écran semi-transparent. Le "workbench" est composé d'une surface de projection horizontale ou inclinée, à laquelle s'ajoute souvent une surface de projection verticale. Il est destiné à une utilisation individuelle. Le "powerwall" est composé d'une ou plusieurs projections alignées sur une surface qui peut être plane ou semi-cylindrique. La CAVE [40] est un cube suffisamment grand pour que plusieurs personnes se tiennent à l'intérieur et dont trois à six faces sont des surfaces de projection. Ces trois dispositifs sont habituellement combinés à un rendu stéréoscopique passif ou actif et à un suivi des mouvements de la tête.

Les écrans volumétriques permettent de se passer de lunettes stéréoscopiques en affichant simultanément tous les points de vue autour d'un objet ou d'une scène 3D. Cependant ils restent limités à des projections de petite taille.

Finalement, les utilisateurs peuvent être équipés de casques, appelés Head-mounted Display, présentant généralement un écran pour chaque œil. Les utilisateurs sont alors totalement coupés de l'environnement réel mais la résolution d'affichage est plus faible que celle des dispositifs basés sur des projections vidéos.

### **Interaction**

L'interaction 3D [23] avec des environnements virtuels est généralement décomposée en trois catégories : la sélection et manipulation d'objets, la navigation et le contrôle d'applications.

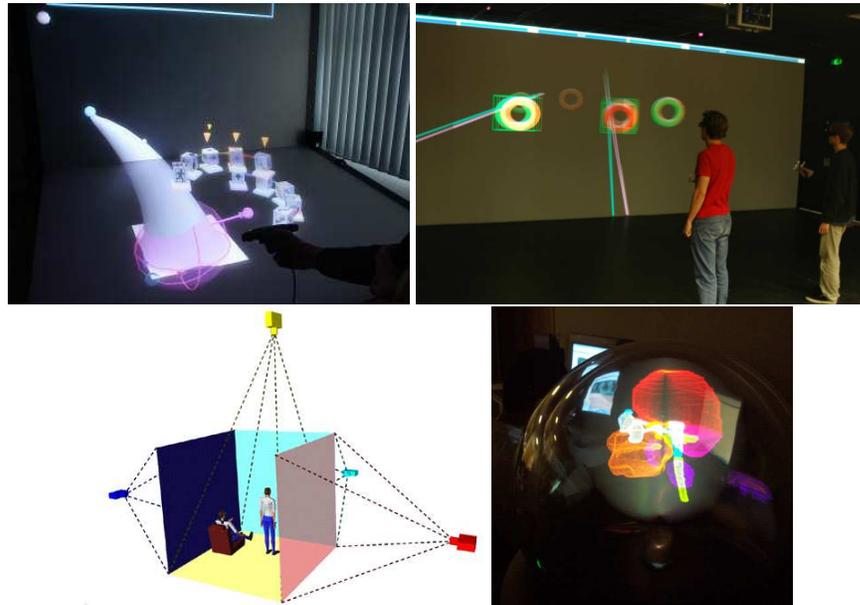


FIG. 1.14 – Dispositif d’affichage : Workbench (tiré de [52]), Powerwall, CAVE, Affichage volumétrique

**Sélection et Manipulation d’objets** Parmi les techniques d’interaction existantes [21], la technique de sélection la plus commune est le rayon virtuel [22]. Un rayon, dont la position de la base et l’orientation sont associées au dispositif d’interaction, se projette dans l’environnement virtuel et permet de sélectionner des objets 3D en les pointant. Cette technique présente certaines limitations auxquelles différents travaux essaient de remédier : accès aux objets occultés, collaboration, précision à une certaine distance. Une fois un objet sélectionné d’autres techniques telles que HOMER [22] permettent de le manipuler avec 6DDL et plus de précision surtout pour les objets éloignés.

Les dispositifs utilisés pour la sélection et la manipulation d’objets possèdent le plus souvent 6DDL et peuvent se répartir en dispositifs fixes et mobiles. Les dispositifs fixes comprennent des dispositifs avec retour haptique actif tels que le Phantom [87], le Spidar [19]. Ces derniers sont plutôt destinés à une utilisation individuelle. D’autres dispositifs favorisent l’interaction collaborative, comme le CAT [60] qui permet en outre de combiner des manipulations 6DDL et des interactions 2D.

Les dispositifs libres sont essentiellement des baguettes munies de boutons poussoirs ou joysticks, ou des gants de données. Leurs position et orientation sont captées grâce à différents systèmes de suivi, dont deux exemples sont donnés sur la figure 1.15. On trouve ainsi du suivi infrarouge utilisant une ou plusieurs caméras et des marqueurs composés de diodes infrarouges ou d’éléments réfléchissants. Ce suivi peut également s’effectuer à l’aide de capteurs et d’émetteurs électromagnétiques ou ultrasons. Chacun

de ces systèmes a ses avantages et inconvénients notamment en termes d'occultations, de latence, de volume de suivi, de précision.



FIG. 1.15 – Suivi vidéo infrarouge avec une caméra et des marqueurs fixés sur un dispositif d'interaction de type "baguette", et tracking électromagnétique.

**Navigation** On peut séparer les techniques de navigation en techniques "réalistes" et "non réalistes". Les techniques "réalistes" comprennent par exemple les techniques de marche virtuelle, dans lesquelles l'utilisateur avance "normalement" dans l'environnement virtuel. Celles-ci peuvent s'effectuer avec des dispositifs de marche réelle. Les utilisateurs sont alors équipés de HMD, se déplacent physiquement dans la pièce et peuvent éventuellement être redirigés par des éléments graphiques afin de simuler un espace plus large [35]. D'autres dispositifs permettent de marcher tout sans de déplacer physiquement soit par les mouvements de pieds, soit par les mouvements du corps ou de la tête. Selon Usoh et al. [116] la marche réelle reste cependant plus efficace que la marche sur place.

Les techniques "non réalistes" sont des techniques de positionnement de point de vue [20]. On trouve ainsi le "vol" dans lequel l'utilisateur se déplace et s'oriente dans les trois dimensions, mais également des techniques qui privilégient l'efficacité de la navigation comme le Go-To ou Navidget [59].

**Contrôle d'applications** L'interaction dans les environnements virtuels nécessite finalement des contrôles de plus haut niveau, sur les paramètres de l'environnement, le chargement d'objets 3D, c'est à dire tout ce qui nécessiterait des menus ou formulaires en 2D. Composants graphiques adaptés aux environnements virtuels, par exemple patchs 3D ou encore menus 3D [24] [53].



## Chapitre 2

# Instruments multi-processus

*Samples have a certain reality. It doesn't just take the sound, it takes the whole way it was recorded. There's a lot of things in the sample, just like when you take a picture—it's got a lot more levels than say, the kick-drum of the drum machine, I think.*

Amon Tobin

Ce chapitre présente les instruments multi-processus. Nous proposons notamment un nouveau modèle d'instrument multi-processus : le live-looping hiérarchique. Nous donnons finalement les problématiques liées à l'implémentation de ces instruments dans des environnements virtuels.

### 2.1 Instrument multi-processus

#### 2.1.1 Définition

Comme nous l'avons décrit dans le chapitre 1, l'évolution des technologies a permis aux musiciens de combiner les rôles d'instrumentiste et de chef d'orchestre. En particulier, les instruments numériques permettent de gérer plusieurs processus sonores, tels que nous les avons définis dans la section 1.1.2, simultanément à partir d'un même contrôleur matériel ou d'une même interface graphique.

On parle alors d'instruments multi-processus [72], qui peuvent être schématisés comme sur la figure 2.1.

Les performances de musiques électroniques reposent généralement sur ces instruments. Les "laptop musicians" utilisent par exemple uniquement un ordinateur portable sur lequel s'exécute le plus souvent un instrument multi-processus. Ces instruments sont également utilisés par des instrumentistes s'accompagnant de multiples séquences musicales.

Les processus sonores peuvent être plus ou moins indépendants, et nécessitent plus ou moins de contrôle de la part du musicien. Ainsi une piste audio peut être suffisamment intéressante musicalement pour se dérouler sans autre intervention

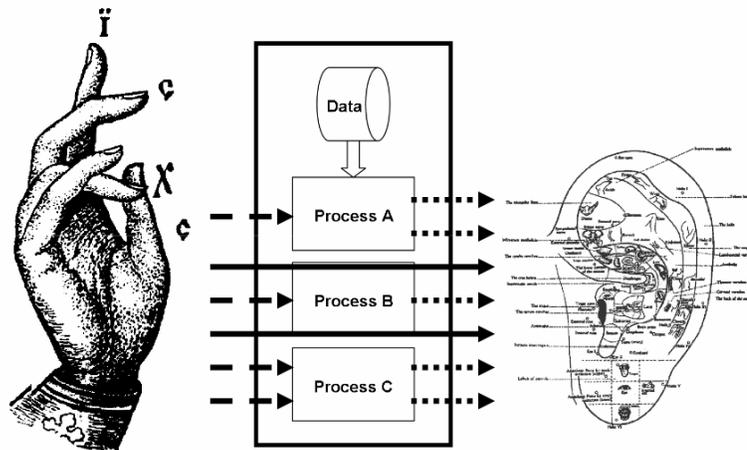


FIG. 2.1 – Illustration d’un instrument multi-processus, tirée de [72]. L’utilisateur intervient sur les différents processus sonores, certains s’appuyant sur des données prédéfinies. Ces processus sonores produisent des flux audio qui sont convertis pour produire le retour auditif.

qu’un déclenchement initial, tandis qu’un processus de synthèse additive nécessitera le déclenchement de chacune des notes à jouer.

La structure des instruments multi-processus peut se modifier au cours d’une performance musicale :

– Ajout/Suppression :

Les processus peuvent être créés/ajoutés et supprimés. Dans le cas des instruments de live-looping, que nous décrivons dans la section 2.2.2, des sons joués en boucle sont ajoutés et supprimés au cours d’une performance.

– Connexions/Déconnexions :

Les flux audio ou de contrôles en sortie des processus, ou entre leurs composants, peuvent être redirigés vers d’autres processus. Un instrument tel que la Reactable permet ainsi de modifier les connexions des composants des processus sonores de manière dynamique, par exemple en connectant un composant de contrôle à un composant de synthèse puis à un autre afin de modifier le résultat sonore à partir des mêmes notes.

Deux principales configurations logicielles d’instruments multi-processus existent. Les processus sonores peuvent être regroupés dans un seul logiciel centralisant le contrôle et l’agencement temporel comme présenté sur la figure 2.2.

Tous les processus sonores peuvent être internes à l’application, ou prendre la forme de “plugins”, composants logiciels externes hébergés par l’application principale. L’autre approche possible est une organisation modulaire avec plusieurs applications, communiquant entre elles ou non, synchronisées ou non. Par exemple, le serveur



FIG. 2.2 – Logiciel LMMS. Les différents processus sonores sont regroupés dans un seul logiciel.

de son Jack Audio Connection Kit <sup>1</sup> permet d'interconnecter des applications musicales en routant les flux audio et MIDI (contrôle). Un exemple est donné sur la figure 2.3.

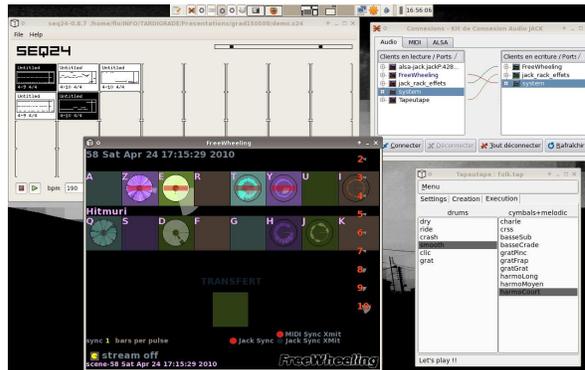


FIG. 2.3 – Plusieurs logiciels interconnectés grâce au serveur Jack.

<sup>1</sup><http://jackaudio.org/>

### 2.1.2 Limitations des contrôleurs matériels

Plus de processus sonores signifie plus de paramètres à gérer simultanément. Ainsi il est important de pouvoir sélectionner les processus et paramètres à manipuler parmi une grande quantité de paramètres. De plus, ces processus s'exécutant simultanément, et n'étant pas forcément toujours audibles, e.g. une piste musicale dont le volume est coupé, il est essentiel pour le musicien de pouvoir connaître l'état de chacun d'entre eux à tout instant. Cet état peut être simple (position de la tête de lecture) ou complexe (automation de plusieurs paramètres, structures/composants). Finalement, le nombre de processus, et donc de paramètres peut très bien varier au cours d'une performance musicale, soit selon une partition écrite, soit selon des modifications structurelles de l'instrument.

Les contrôleurs matériels, tel que celui présenté sur la figure 2.4 (gauche) montrent alors leurs limites, s'ils ne sont pas accompagnés d'une interface graphique.

Comment en effet donner accès à un nombre potentiellement infini, et qui plus est variable, de processus à l'aide d'un ensemble fini et fixe de capteurs ?

Le musicien doit choisir un certain nombre de paramètres qu'il veut contrôler, et éventuellement utiliser des presets pour contrôler les autres, limitant ainsi les possibilités d'interaction et figeant sa performance. De plus l'accès à ces paramètres est souvent ralenti par leur organisation sous forme de menus, car ceux-ci déconnectent l'utilisateur de la tâche principale et impliquent donc une surcharge cognitive.

Comment visualiser l'état de ces processus à l'aide de composants simples tels que des diodes lumineuses ?

Les informations se limitent donc souvent à l'activation ou à l'amplitude (vu-mètre) du signal des processus sonores. La structure de l'instrument et des processus sonores ne peut pas être affichée.



FIG. 2.4 – Contrôleur matériel M-Audio et Interface graphique du logiciel Freewheeling

### 2.1.3 Avantages de l'interaction graphique

Les interfaces graphiques, comme celle du logiciel Freewheeling<sup>2</sup> que l'on peut voir sur la partie droite de la figure 2.4, vont présenter dans le cas des instruments multi-processus un certain nombre d'avantages, auxquels s'ajouteront ceux des environnements virtuels comme nous le verrons dans la section 2.3.

#### Sélection

Le musicien a à sa disposition un nombre potentiellement infini de composants graphiques, surtout si la navigation dans l'espace graphique est possible. Il peut directement sélectionner des processus et des paramètres à manipuler à l'aide de techniques efficaces de pointage. L'apparence (couleurs/forme/labels/...) des composants graphiques permet de rapidement les identifier, par exemple grâce aux techniques issues du domaine de la visualisation d'informations.

#### Manipulation

Chaque composant graphique possède plusieurs paramètres graphiques modifiables, parfois simultanément. Par ailleurs, de nombreuses techniques de manipulation existent, comme nous avons pu le voir dans le chapitre 1. Ces techniques peuvent imiter des manipulations réalistes : potentiomètres virtuels, modèles physiques, dessin/sculpture. Mais il est également possible de dépasser les contraintes physiques. Par exemple, un paramètre pourra être modifié suivant plusieurs échelles représentées graphiquement. Le musicien pourra également sélectionner et manipuler des ensembles de composants graphiques simultanément, dans la limite des degrés de libertés du dispositif d'interaction utilisé. Cette limite est par ailleurs levée par les possibilités de sélection, ou mappings entre le dispositif d'interaction et les objets graphiques, apportées par l'interface graphique.

#### Visualisation

Une multitude de paramètres graphiques peuvent être utilisés, aussi bien des paramètres spatiaux (position sur 2 ou 3 axes, orientation sur 1 ou 3 axes,...), que des paramètres d'apparence (couleur, opacité, texture,...), que des paramètres de forme (formes multi-dimensionnelles, déformation simple, formes paramétriques,...). Ces paramètres peuvent de plus être combinés afin qu'un composant graphique représente plusieurs paramètres d'un même processus sonore. Cette visualisation permet au musicien de connaître l'état de chaque processus sonore (position de la tête de lecture, spectrogramme, ...) avec la précision souhaitée.

---

<sup>2</sup><http://freewheeling.sourceforge.net/>

## 2.2 Modèle d'instrument multi-processus : Live-looping hiérarchique

Dans cette section, nous présentons un modèle d'instrument multi-processus, le *live-looping hiérarchique*, que nous souhaitons mettre en œuvre dans un environnement virtuel.

### 2.2.1 Live-looping

#### Principe

Le live-looping est une technique musicale qui consiste à enregistrer un flux audio ou de contrôle puis à lire cet enregistrement en boucle. L'opération peut être répétée plusieurs fois afin d'empiler des boucles. Celles-ci peuvent être synchronisées de plusieurs façons afin de permettre de construire des rythmes sans que les différentes boucles ne se décalent. Généralement leurs longueurs sont modifiées afin de correspondre à des multiples de la longueur d'une pulsation qui peut être la première boucle enregistrée ou une pulsation externe (métronome). Elles peuvent également être toutes démarrées simultanément lorsque seule l'une d'elles est déclenchée. Certains dispositifs de live-looping permettent ensuite de manipuler les boucles indépendamment, ce sont alors des instruments multi-processus.

L'utilisation de cette technique est de plus en plus courante chez les instrumentistes et chanteurs solo. Cela leur permet de se créer rapidement et facilement un accompagnement, et notamment avec très peu d'instruments (une guitare peut être frappée pour constituer une base rythmique, puis jouée normalement...). De nombreux musiciens électroniques s'en servent également. Bien que cette technique soit essentiellement destinée au jeu "live" (improvisé ou non), elle n'en reste pas moins intéressante pour la composition, puisqu'elle permet de rapidement tester des arrangements.

Dans le cadre de la musique électronique, le live-looping présente plusieurs avantages par rapport aux autres techniques de jeu.

- Tout d'abord contrairement aux séquences MIDI couramment utilisées, le live-looping n'impose pas de quantification temporelle. Les rythmes peuvent être enregistrés avec les légères imperfections qui constituent un jeu naturel, contrairement à des séquenceurs qui aligneraient exactement les notes sur des subdivisions des temps.
- Différentes sources peuvent être enregistrées. Ceci inclut n'importe quelle source audio (instrumentiste, synthèse) mais également des événements musicaux (notes, gestes, modifications de paramètres). Ces libertés au niveau de la temporalité et des sources permettent d'utiliser le live-looping pour n'importe quel genre musical, avec n'importe quelle signature rythmique.
- Dans un contexte de performance, il est possible de mélanger des boucles pré-enregistrées et des boucles improvisées.
- La superposition de boucles très courtes permet de créer de nouvelles textures sonores facilement.

### Historique

Les origines du live-looping remontent aux expérimentations menées par des compositeurs tels que Terry Riley ou Steve Reich qui, grâce à des bandes magnétiques, composent de la musique minimaliste en superposant des boucles sonores, en les étirant, en les inversant. Cette idée est ensuite mise à profit par différents constructeurs, donnant naissance à des dispositifs électroniques sous forme de racks comme le Gibson Echoplex puis de pédaliers comme le Digitech Jamman ou le Boss RC20. Les opérations alors ajoutées sont l'enregistrement sur une seule boucle de plusieurs couches (overdub), la multiplication de la durée d'une boucle, la synchronisation par le protocole MIDI, des contrôles plus directs et plus efficaces. Suivent des fonctionnalités telles que la gestion indépendante de plusieurs boucles, et l'enregistrement d'ensemble de boucles synchronisées sous forme de "scènes".

Finalement des logiciels de live-looping sont développés. On peut notamment citer Sooperlooper<sup>3</sup> et Freewheeling<sup>4</sup>, ce dernier étant représenté sur la figure 2.5. L'interaction avec ces instruments multi-processus est facilitée par l'utilisation d'interfaces graphiques. Cela permet d'ajouter la gestion de pulsations multiples pour synchroniser les boucles par groupes, la visualisation des boucles qui s'avère très utile lorsque leur nombre devient important, la sélection et manipulation simultanée de plusieurs boucles, un nombre illimité de boucles, des scènes plus faciles à sélectionner. Certains de ces logiciels donnent aussi la possibilité d'avoir des sorties audio séparées, une pour chaque boucle, afin d'appliquer des effets audionumériques.



FIG. 2.5 – Logiciel de live-looping Freewheeling. 5 boucles sur 7 sont activées, la pulsation de synchronisation est affichée en haut à droite. Les signaux des boucles sont représentés de manière circulaire.

<sup>3</sup><http://www.essej.net/sooperlooper/>

<sup>4</sup><http://freewheeling.sourceforge.net/>

### Audio/Contrôle

Deux types de live-looping existent. Le live-looping audio est la technique la plus commune. Les boucles contiennent dans ce cas un flux audio échantillonné. Mais on trouve également du live-looping de contrôle. Dans ce cas, les boucles contiennent des événements de contrôles qui ont une date et un ensemble de paramètres. Le logiciel LiveLoop<sup>5</sup> permet ainsi de boucler des séquences de messages MIDI. Les avantages de ce live-looping de contrôle sont nombreux et tiennent surtout à ce que les données enregistrées sont des événements indépendants et séparés.

- Tout d’abord chaque événement peut être modifié indépendamment. Cela permet d’ajouter, de supprimer des événements mais également d’en modifier les paramètres. Par exemple, il est possible de modifier les vitesses d’une série de notes afin de transformer un rythme. Cette opération serait par contre plus compliquée à réaliser avec du live-looping audio car il faudrait extraire les différentes notes du signal, puis les réinsérer.
- L’étirement temporel et les autres modifications temporelles (répétitions, ...) sont plus simples à effectuer que dans le cas du live-looping audio puisqu’il suffit de modifier les dates des événements.
- Les contrôles discrets et continus (effets) sont séparés. Ils peuvent donc être contrôlés séparément. Cela peut permettre de modifier ou de supprimer seulement les effets appliqués sur une séquence.
- Finalement, les événements de contrôle étant séparés des processus sonores qu’ils modifient, ils peuvent facilement être réaffectés. Par exemple il est possible de modifier le canal MIDI d’une note afin de déclencher un autre son.

Cependant, le live-looping de contrôle est plus compliqué à mettre en œuvre. En effet, les connexions entre les séquences et les processus de synthèse doivent être gérées (sauvegardées, restaurées, ...), ce qui peut se révéler complexe en fonction du nombre de processus et de séquences utilisés. De plus, une séquence est en quelque sorte liée à un processus de synthèse, et si celui-ci est modifié, la boucle originale est altérée. Une boucle audio est par contre indépendante, elle ne changera pas si les paramètres de la source audio sont modifiés.

### Nouveaux instruments de live-looping

Parmi les nouveaux instruments qui s’appuient sur la technique du live-looping, on trouve notamment Beatbugs [132] et Fijuu [98], représentés sur la figure 2.6. Les Beatbugs sont des petits dispositifs matériels qui permettent d’enregistrer des rythmes simples et de les rejouer. Plusieurs dispositifs peuvent être synchronisés, et les boucles peuvent être modifiées. Cependant, un beatbug ne peut gérer qu’une boucle à la fois.

Fijuu est un instrument virtuel 3D. Il associe des formes 3D, disposées au milieu d’une scène comportant des pistes circulaires et une tête de lecture, à différents processus de synthèse sonore. L’interaction s’effectue grâce à une manette de jeu. L’utilisateur peut ainsi sélectionner et déformer ces objets 3D, contrôlant alors différents paramètres de synthèse. Il peut également couper le volume de synthèse. Il est ensuite possible d’enregistrer une boucle audio de ces manipulations. Les boucles enregistrées

<sup>5</sup><http://code.google.com/p/liveloop/>

sont assignées à des pistes circulaires, dont le diamètre correspond au volume de lecture. Ceci permet de visualiser les boucles et d’y accéder simplement afin de les altérer en opérant des coupures de volume. En plus des modifications par boucle, il est possible d’appliquer des effets globaux qui sont répercutés sur l’affichage de la scène 3D.

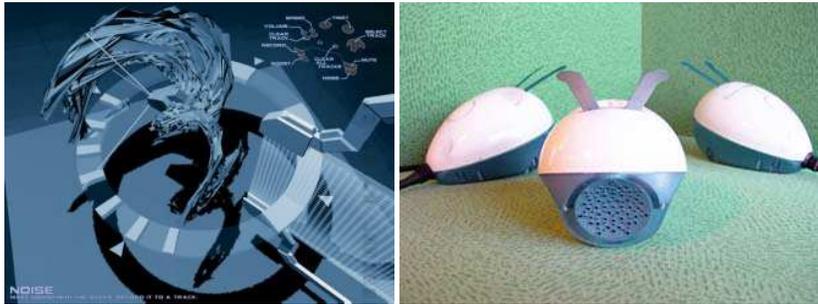


FIG. 2.6 – Fijuu et Beatbugs

### 2.2.2 Live-looping Hiérarchique

La technique du live-looping telle qu’appliquée par les instruments existants impose cependant des limites pour la création de structures musicales. En effet, les boucles peuvent seulement être empilées. Il est compliqué d’organiser les boucles sous forme de groupes, et d’accéder à la fois à ces groupes et aux boucles qui les constituent. Dans le cas du live-looping audio, les modifications des boucles sont de plus très limitées, et le regroupement de plusieurs boucles (overdub) reste figé. Le live-looping de contrôle permet plus de flexibilité mais au prix de la difficulté de gestion des connexions avec les processus de synthèse.

Nous proposons donc de combiner ces deux types de live-looping en ajoutant une structure arborescente s’inspirant de celle mise au point par Marczak [85]. Ce dernier définit un système hiérarchisé de boîtes de contrôle musical, qui couvre différentes étapes d’une composition électroacoustique, de la création de l’instrument (lutherie numérique) à la performance.

Notre technique, constituant un nouveau modèle d’instrument multi-processus, est nommée *live-looping hiérarchique*. Elle est décrite dans les sous-sections qui suivent.

#### Principe

Le *live-looping hiérarchique* repose sur la notion d’*arbres de live-looping*. Les nœuds de ces arbres sont définis ainsi :

$$\text{noeud}\{\text{contenu}, \text{mode}, \text{effets}[], \text{enfants}[]\}$$

où :

- "contenu" est un contenu musical
- "mode" est le mode de lecture du contenu lorsqu'il est déclenché. Dans le mode "trigger", le contenu est lu en entier. Dans le mode "normal", le contenu est lu jusqu'à sa fin ou réception d'un évènement d'arrêt. Finalement, dans le mode "loop", le contenu est lu en boucle jusqu'à réception d'un évènement d'arrêt.
- "effets" une suite d'effets audionumériques et d'effets de contrôle appliqués sur le contenu.
- "enfants" est une liste d'enfants

Les feuilles et les nœuds de plus haut niveau se distinguent par leur contenu, leurs effets et évidemment l'existence ou non d'enfants.

Les caractéristiques des feuilles sont les suivantes :

- La liste d'enfants est vide.
- Dans notre implémentation actuelle, le contenu musical des feuilles est un composant de synthèse sonore qui permet de lire un ensemble de fichiers audio stéréo constituant un instrument multi-échantillonné soit de manière simple (lecture du son en entier, en suivant le mode de lecture du contenu) avec un paramètre de déclenchement et d'arrêt, soit par synthèse granulaire, avec un paramètre de position de la tête de lecture des grains. Il serait cependant possible de remplacer ce contenu par un composant de synthèse additive par exemple.
- Le mode de lecture ne peut pas être "loop".
- La suite d'effets est uniquement composée d'effets audionumérique, qui sont dans notre implémentation actuelle : la modification de hauteur, le volume, la distorsion et la réverbération.

Pour les nœuds de plus haut niveau (nœuds parents) :

- La liste d'enfants n'est pas vide
- Le contenu est une liste de séquences d'évènements de contrôle (une séquence par enfant), évènements enregistrés à partir du flux de contrôle provenant du musicien. Chaque séquence contient plusieurs pistes, une pour chaque effet et une pour les déclenchements de contenu. Les durées des pistes d'effets sont ajustées sur des multiples de la durée de la piste de déclenchement. Les séquences du contenu sont synchronisées entre elles en alignant leurs durées sur des multiples de celle de la séquence la plus ancienne. Ces séquences correspondent aux boucles du live-looping classique.
- Le mode de lecture est "loop" par défaut mais il peut être modifié en "trigger" ou "normal".
- La suite d'effets est composée des mêmes effets audionumériques que les feuilles, auxquels s'ajoutent des effets de contrôle qui s'appliquent à toutes les séquences du contenu.

Les effets audio s'appliquent sur tout le flux audio venant des nœuds enfants, comme représenté sur la figure 2.7. Ce flux audio remonte donc depuis le contenu audio des feuilles jusqu'à la racine de l'arbre, en passant par les effets de tous les nœuds parents. A l'inverse, le flux de contrôle descend dans l'arbre. En effet, le déclenchement du contenu de la racine va entraîner la lecture des évènements des

nœuds enfants, c'est à dire en partie des déclenchements de séquences. Le contrôle redescend ainsi jusqu'au déclenchement du contenu musical des feuilles. Cependant, il est également possible d'accéder directement aux différents paramètres des effets et des contenus. A ce flux descendant s'ajoutent donc des flux de contrôle directs vers chaque nœud.

Finalement, le principe des scènes utilisé dans le live-looping classique est également présent dans le *live-looping hiérarchique*. Les scènes sont composées d'un ou de plusieurs arbres de live-looping, qui peuvent donc ne pas être synchronisés. Ainsi chaque scène peut s'appuyer sur des contenus audio (dans les feuilles) différents, ou bien sur les mêmes mais développés de manière différente.

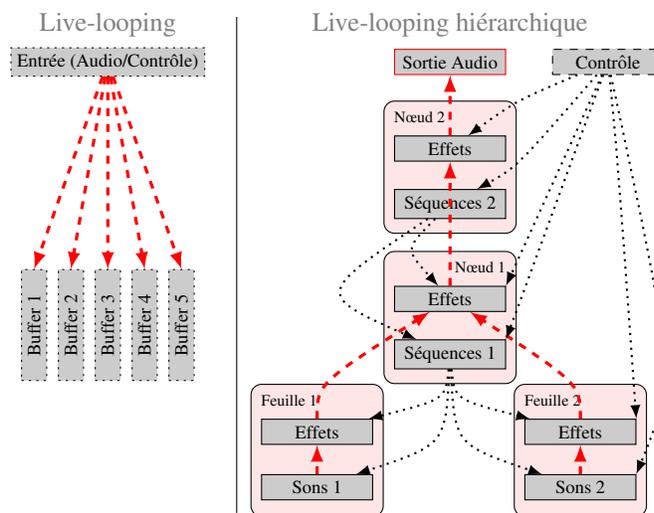


FIG. 2.7 – Flots de données du Live-looping (à gauche) et Flots de données de contrôle (en noir) et audio (en rouge) du Live-looping hiérarchique. Le flot de contrôle direct vers la Feuille 1 n'est pas représenté pour ne pas surcharger le schéma, mais il est possible d'accéder directement à tous les nœuds.

### Opérations sur les nœuds

Les nœuds peuvent être manipulés grâce à deux opérations. Leur contenu, que ce soit le composant de synthèse des feuilles ou les séquences des nœuds parents, peut tout d'abord être déclenché et arrêté (suivant le mode de lecture). Suivant le composant de synthèse utilisé comme contenu des feuilles, d'autres paramètres peuvent également être contrôlés. Toutes ces manipulations sont des événements de contrôle enregistrables dans les séquences des nœuds parents.

La deuxième opération disponible concerne les effets audionumériques, auxquels s'ajoutent les effets de contrôle dans les nœuds parents. Leurs paramètres peuvent être

modifiés, chaque modification correspondant à un évènement de contrôle enregistrable dans les séquences.

### Opérations sur les arbres

Quatre opérations permettent de manipuler les *arbres de live-looping*. Elles sont représentées sur la figure 2.8.

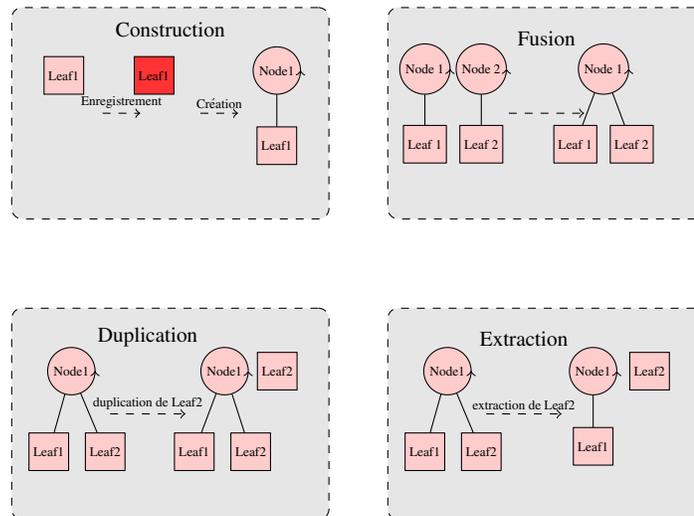


FIG. 2.8 – Opérations du live-looping hiérarchique.

L'opération de construction est l'opération de base du live-looping hiérarchique. Elle permet d'enregistrer une boucle d'évènements d'un nœud, et de créer un nœud parent qui contiendra cette séquence. Elle comprend deux étapes. La première étape est le déclenchement de l'enregistrement des évènements d'un nœud. Lors de la deuxième étape, l'enregistrement est arrêté, un nouveau nœud est créé avec pour contenu les séquences enregistrées et devient le parent du premier nœud. La séquence enregistrée est alors jouée en boucle, comme c'est le cas pour le live-looping classique. Si le nœud sélectionné pour l'opération possède déjà un parent, l'opération construire réenregistre seulement les séquences des paramètres modifiés. Ceci permet d'enregistrer les évènements en plusieurs fois, par exemple dans un premier temps les déclenchements du contenu, puis les modifications des effets.

L'opération de fusion consiste à lier deux *arbres de live-looping* en fusionnant deux nœuds parents. Elle n'est par contre pas autorisée si ces nœuds ont tous les deux des parents, car cela entraînerait des conflits, notamment au niveau des flux de contrôle descendants et des flux audio remontant. Lors de la fusion, les séquences des contenus sont regroupées et elles sont synchronisées sur des multiples de la séquence la plus ancienne, comme présenté sur la figure 2.9, et conformément au contexte musical du live-looping (empilement progressif des boucles). Ainsi le nœud parent synchronise

tous ses enfants ensemble, ce qui permet par exemple de créer des groupes rythmiques.

Les opérations de construction et de fusion peuvent être réalisées en même temps sur plusieurs nœuds. Les parents créés sont tous immédiatement fusionnés.

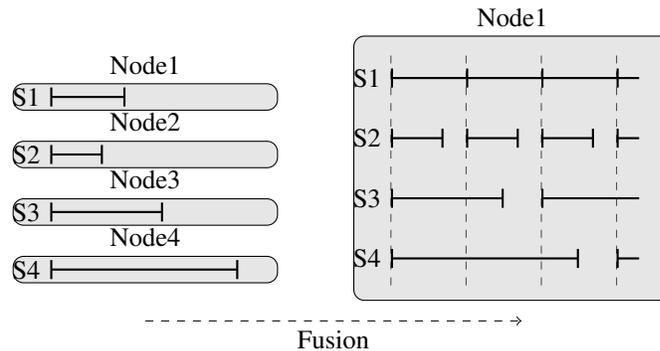


FIG. 2.9 – Synchronisation de séquences lors de la fusion de plusieurs nœuds. Ici la séquence la plus ancienne est S1. Les autres séquences sont donc synchronisées à des multiples de S1.

L'opération de duplication est très utile car elle permet de contourner le problème lié au live-looping de contrôle, et qui se retrouve dans le *live-looping hiérarchique*. Lorsqu'un nœud est utilisé dans un arbre, il ne peut faire partie d'un deuxième arbre (toujours pour éviter des conflits entre les flux de contrôle et entre les flux audio). Ainsi, les contenus des feuilles ne sont utilisables qu'une fois. L'opération de duplication permet de copier un nœud et ses enfants (donc un arbre), afin de les réutiliser. Les paramètres des nœuds de l'arbre peuvent alors être modifiés, et cet arbre peut être soit ajouté à l'arbre précédent, soit constituer la base d'un nouvel arbre. Les mêmes contenus peuvent donc être réutilisés à volonté.

Finalement, l'opération d'extraction permet d'extraire un nœud et ses enfants d'un arbre. La séquence correspondante est effacée du nœud parent, l'arbre (ou la feuille) obtenue devient une racine (ou reste une feuille). Cette opération, appliquée sur une racine, c'est à dire un nœud parent créé avec l'opération de construction, permet également d'arrêter la lecture en boucle du contenu.

### Avantages

Le *live-looping hiérarchique* possède un certain nombre d'avantages par rapport au live-looping classique.

- Tout d'abord, il permet de construire facilement des structures musicales variées et complexes en combinant les opérations de construction et de fusion, là où le live-looping classique permet seulement d'empiler des boucles. Deux exemples d'*arbres de live-looping* sont donnés dans la figure 2.10.
- Chaque nœud reste accessible indépendamment, ce qui va permettre de les modifier par groupes (en modifiant les effets d'un nœud parent on modifie tous ses

enfants) ou individuellement (déclenchement/réenregistrement des séquences). Il est donc possible de combiner les avantages de l'overdub du live-looping audio (fusion de boucles synchronisées) et du live-looping de contrôle (accès indépendant aux éléments enregistrés).

- Cette technique, de par l'organisation même des *arbres de live-looping*, s'adapte à différents niveaux d'expertise. En effet, les feuilles contiennent des sons simples, bruts, et donc nécessitent une certaine pratique afin de pouvoir produire un résultat musical intéressant. D'un autre côté, lorsque l'on remonte dans un arbre, le déclenchement des contenus des nœuds va entraîner la lecture de séquences de plus en plus complexes.
- Finalement, le jeu collaboratif offre beaucoup de perspectives. Comme dans le live-looping traditionnel, les musiciens peuvent construire chacun des boucles puis les synchroniser/fusionner. Mais les arbres créés par un musicien peuvent également être dupliqués pour être modifiés simultanément par un autre. Les musiciens peuvent aussi se répartir les tâches, l'un effectuant des opérations sur les nœuds (déclenchements, effets) alors que l'autre effectue des opérations sur les arbres.

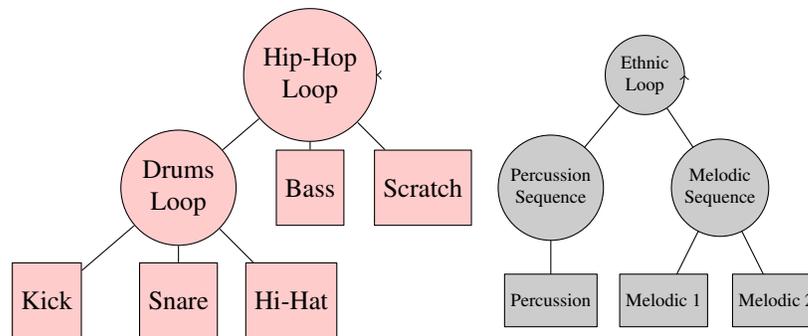


FIG. 2.10 – Deux exemples d'arbre de live-looping.

## 2.3 Environnements virtuels immersifs et instruments multi-processus

### 2.3.1 Environnements virtuels

Pour certains instruments multi-processus l'utilisation d'interfaces graphiques 2D peut s'avérer complexe, voire impossible. Par exemple, plusieurs contraintes sont imposées dans le cas du *live-looping hiérarchique*.

- La visualisation et la manipulation des arbres ne doit pas perturber la manipulation et la visualisation de chacun des nœuds.
- De nombreux paramètres doivent pouvoir être manipulés sur chaque nœud aussi bien pour le contenu que pour les effets.
- Il doit finalement être possible de manipuler des structures dynamiques, et de naviguer dans la liste des scènes de *live-looping*, qui contiennent toutes plusieurs arbres.

Dans de telles conditions, une interface 2D pourrait se trouver surchargée et inutilisable.

Les environnements virtuels et l'interaction 3D peuvent faciliter la mise en œuvre de ces instruments plus complexes et également apporter d'autres avantages.

#### Visualisation

Ils donnent accès à des paramètres graphiques supplémentaires par rapport aux composants graphiques 2D. On trouve par exemple des effets de transparence, d'éclairage, d'ombres, le relief des textures, les formes 3D, des axes de rotation et de translation supplémentaires. Pour les instruments multi-processus, ceci signifie plus d'informations sur chaque processus sonore, et donc une interaction facilitée.

L'agencement spatial en 3 dimensions permet d'afficher un plus grand nombre de processus, structurés ou non, qui vont être identifiés aussi efficacement [36] que pour une interface 2D [37]. Les arbres du *live-looping hiérarchique* ne seront donc pas limités à une disposition sur un plan.

#### Navigation

De même les techniques de navigation 3D existantes, que nous avons décrites dans le chapitre 1, vont permettre de parcourir les processus ou groupements de processus agencés dans l'espace 3D. Par exemple, cela pourra nous permettre de naviguer dans les scènes du *live-looping hiérarchique*. Cette navigation pourra également être utilisée comme métaphore du parcours temporel des processus.

#### Manipulation

Les nombreuses techniques d'interaction 3D existantes peuvent être mises à profit pour les instruments multi-processus. Elles permettent notamment de sélectionner des objets 3D de manière efficace et de manipuler les nombreux paramètres graphiques

apportés par les environnements virtuels. Elles doivent cependant être adaptées au contrôle musical.

### **Immersion**

Finalement, la perception de l'instrument multi-processus et l'interaction avec ce dernier sont facilitées par l'immersion, que nous avons décrite dans le chapitre 1 généralement mise en œuvre pour les environnements virtuels. Dans le cas du live-looping hiérarchique, qui est un instrument principalement destiné à la performance, cette immersion va profiter aussi bien aux musiciens qu'aux spectateurs, comme nous le verrons dans le chapitre 5.

### **2.3.2 Instruments immersifs existants**

Parmi les instruments immersifs existants, on trouve plusieurs instruments "mono-processus", dont quelques-uns sont représentés sur la figure 2.11. On peut citer par exemple, les instruments mis au point par Mäki-Patola et al. [84], à savoir les "Virtual Xylophone", "Virtual Membrane" et "Virtual Air Guitar". On trouve également les travaux menés par Mulder [95] sur la manipulation d'objets virtuels musicaux et l'instrument de sculpture virtuelle [94] qui en résulte. Les instruments Phase [107] et Plumage [69] s'appuient sur la navigation dans des environnements virtuels mais ne contrôlent qu'un seul processus, respectivement une séquence musicale et un processus de synthèse granulaire. Parmi des instruments 3D mono-processus et moins immersifs, on peut citer plusieurs instruments reprenant le principe des jeux de tirs à la première personne, notamment ceux développés par Hamilton [62] [61] ou encore Q3apd [97].

Les instruments multi-processus immersifs peuvent être classés suivant leurs possibilités d'interaction et de visualisation, ce que présente la figure 2.12. Nous donnons également la position de deux instruments non immersifs.

On observe tout d'abord deux niveaux d'immersion dans les applications présentées. Les logiciels Fijuu [98], Ergates [88] et Versum [6] par exemple, sont des instruments multi-processus mais ils sont peu immersifs. Fijuu, dont nous avons parlé dans la section 2.2.2, et Ergates permettent tous deux de sélectionner et d'interagir avec des processus, représentés par des formes 3D et des séquences circulaires, à l'aide de manettes de jeu vidéo. Dans les deux cas, les manipulations, limitées par les dispositifs d'interaction, sont des déformations et des déclenchements. D'un autre côté, on trouve Versum, un séquenceur 3D. Il permet de disposer des entités 3D correspondant à des processus sonores dans un environnement virtuel. La visualisation est plus avancée que dans les logiciels précédents car l'apparence des entités reflète les paramètres sonores. L'interaction se limite cependant au déplacement d'un "acteur", ou micro virtuel, dans cet espace sonore. Le degré de contrôle musical et la liberté laissée au musicien sont donc relativement restreints.

D'autres instruments multi-processus sont beaucoup plus immersifs. Ils utilisent en effet des affichages stéréoscopiques et le suivi des mouvements de la tête. Si on les classe par degré de contrôle musical, on trouve tout d'abord les instruments développés

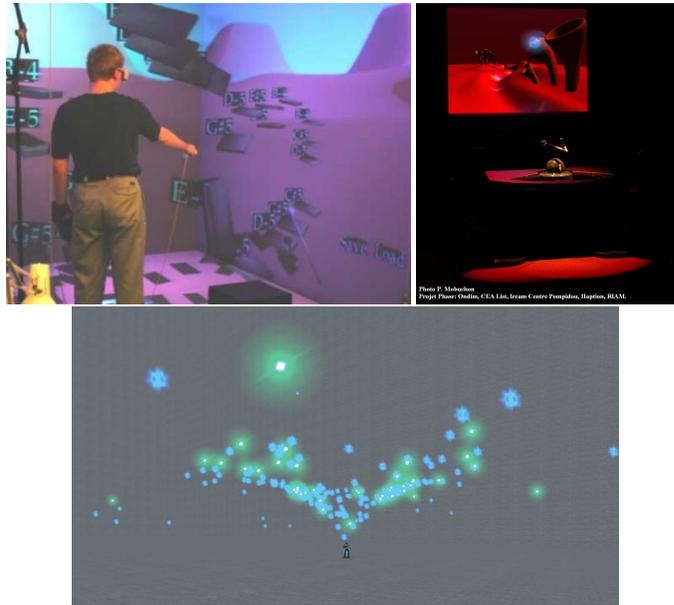


FIG. 2.11 – Virtual Xylophone, Phase et Q3osc

par Wozniowski et al. [135], dont un exemple est l'application d'Écoute active. Cet instrument, représenté sur la figure 2.13 permet à un utilisateur de contrôler les volumes de différents processus représentés par des objets 3D en orientant un capteur de mouvements associé à un avatar.

L'instrument développé par Naef et al. [96] offre un peu plus de possibilités. Plusieurs sources sonores peuvent être spatialisées en positionnant des objets 3D dans l'environnement virtuel, leur volume étant de plus associé à l'orientation des objets.

Finalement, un travail important a d'ailleurs été fourni par Valbom et Marcos [118] sur l'utilisation d'environnements virtuels pour la spatialisation. L'instrument WAVE [117], présenté sur la figure 2.13, imite une surface de contrôle matérielle. Il permet de déclencher différents processus de synthèse (fichiers audio, synthétiseurs, des boucles rythmiques et mélodiques) et d'appliquer des effets MIDI. Ici le degré de contrôle musical est plus élevé, mais il ne permet cependant pas des manipulations de timbre. De plus, comme le reconnaît Valbom [117], le dispositif d'interaction ne permet pas d'actions très précises au niveau temporel, et interdit donc de jouer correctement des rythmes rapides. La visualisation est également très limitée et Valbom remarque que plus d'informations sur les différents processus, par exemple leur spectre ou leur déroulement, aiderait les utilisateurs.

Les limitations en terme d'interaction et de visualisation des instruments multi-processus immersifs existant peuvent expliquer pourquoi ils restent à l'état de prototype dans un laboratoire de recherche, et pourquoi des musiciens ne se les approprient

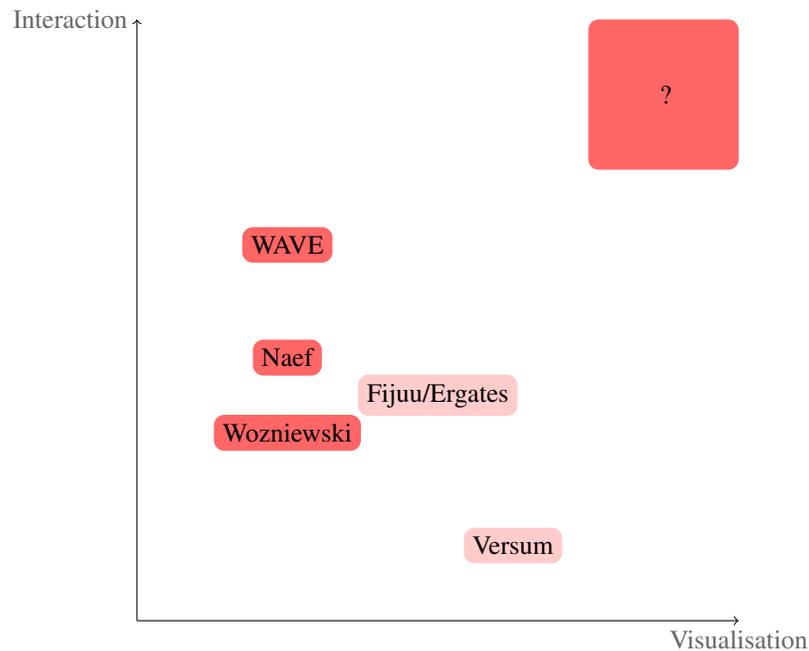


FIG. 2.12 – Instruments 3D multi-processus. Les axes donnent une évaluation des possibilités d’interaction et de visualisation offertes par ces applications. La couleur donne le degré d’immersion (plus l’application est rouge plus elle est immersive).

pas. Notre objectif pour l’implémentation du live-looping hiérarchique, représenté par le point d’interrogation dans la figure 2.12 est d’arriver à combiner les possibilités d’interaction et de visualisation apportés par les environnement virtuel immersifs.

### 2.3.3 Problématiques

Plusieurs problématiques, communes à tous les instruments multi-processus, se posent alors.

#### Représentation des processus

La première question est celle de la représentation des processus sonores. L’objectif ici est de pouvoir facilement identifier ces processus, visualiser leurs paramètres et accéder aux structures Quels composants graphiques de l’environnement virtuel peut-on utiliser ? Comment peut-on les organiser pour une interaction musicale pertinente ? Cette problématique est abordée dans le chapitre 3.

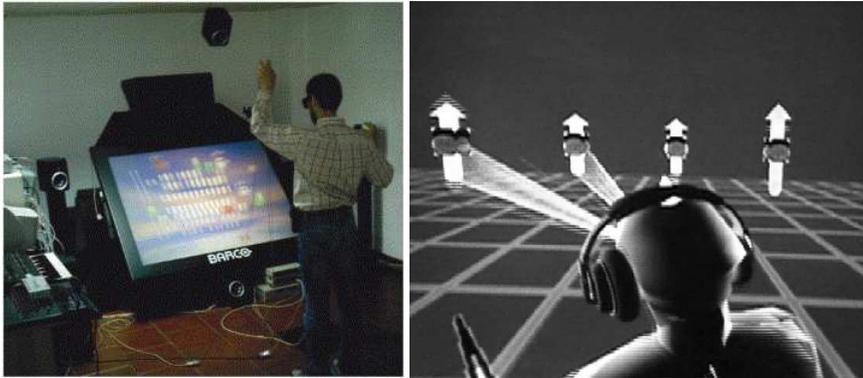


FIG. 2.13 – Instruments immersif multi-processus. WAVE (à gauche) et Application d'Écoute Active (à droite).

### Interaction

La seconde problématique est celle de l'interaction avec ces processus sonores dans les environnements virtuels. Il convient ici de prendre en compte les principes de l'interaction musicale afin de les combiner aux techniques d'interaction 3D. Nous proposons des solutions à cette problématique dans le chapitre 4.

### Mise en œuvre

La dernière question est celle de la mise en œuvre des instruments multi-processus, dans notre cas du *live-looping hiérarchique* : mise en œuvre logicielle s'appuyant sur les solutions proposées pour la représentation et la manipulation des processus sonores, mais également mise en œuvre matérielle pour l'utilisation de cet instrument dans le cadre de performances publiques. Ces problématiques sont traitées dans le chapitre 5.



## Chapitre 3

# Représentation de processus sonores dans des environnements virtuels

*As for the future of electronic music, it seems quite obvious to me that its unique resources guarantee its use, because it has shifted the boundaries of music away from the limitations of the acoustical instrument, of the performer's coordinating capabilities, to the almost infinite limitations of the electronic instrument. The new limitations are the human ones of perception.* Milton Babbitt

Ce chapitre décrit les *widgets réactifs 3D*, éléments graphiques qui permettent de représenter des processus sonores dans un environnement virtuel. Nous abordons également les mappings audiovisuels utilisés pour ces widgets, et la combinaison de ces mappings [10].

### 3.1 Widgets Réactifs 3D

#### 3.1.1 Principe

Nous proposons d'adapter les widgets réactifs décrits par Levin [80] aux environnements virtuels. Un *widget réactif 3D* est un composant graphique 3D simple ou composé correspondant à un processus sonore et dont les paramètres graphiques sont connectés aux paramètres de ce processus sonore. Les connexions peuvent s'effectuer dans le sens de la visualisation (sonore→graphique) et dans le sens du contrôle (graphique→sonore), comme présenté sur la figure 3.1. Il est donc possible de visualiser les paramètres qui varient de manière indépendante (position d'une tête de lecture, spectre ou enveloppe d'un signal audio), et parfois à une fréquence supérieure à la fréquence des gestes musicaux, tout en manipulant des paramètres de synthèse,

d'effets ou encore musicaux. Les *widgets réactifs 3D* reflètent l'état des processus sonores tout en permettant de les modifier. Plusieurs exemples de widgets réactifs 3D sont donnés sur la figure 3.2.

Il est important de noter que les valeurs exactes (chiffrées) des paramètres sonores ne sont pas représentées par les *widgets réactifs 3D*. Bien qu'il soit important d'avoir une précision chiffrée dans le cas d'applications de composition, ce qui implique l'utilisation de composants textuels ou d'axes chiffrés, ce n'est pas le cas pour la performance musicale. En effet, en combinant l'écoute, la visualisation de certaines paramètres sonores (spectre, amplitude) et des échelles de valeurs adaptées pour les paramètres graphiques, le musicien aura suffisamment d'informations pour contrôler l'instrument, et par exemple éviter le clipping du signal sonore (valeurs dépassant 1).

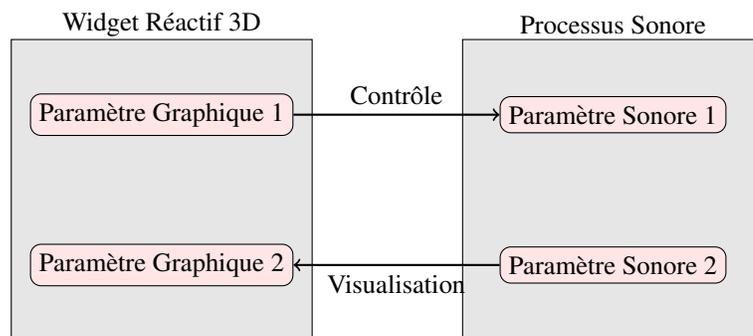


FIG. 3.1 – Widget Réactif 3D : Visualisation du paramètre sonore 2 et contrôle du paramètre sonore 1

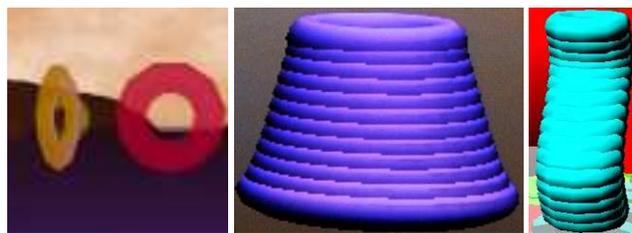


FIG. 3.2 – Exemples de *widgets réactifs 3D* développés au cours de nos travaux. Pour les deux widgets de gauche, les paramètres de couleur et de taille sont connectés à des paramètres sonores. Pour le widget du centre, la forme est utilisée pour représenter le spectre du processus sonore. Le widget de droite utilise à la fois les paramètres de couleur, de transparence, de taille et de forme et d'orientation

### 3.1.2 Avantages pour les instruments multi-processus

Par rapport à d'autres composants graphiques utilisés pour les instruments immersifs tels que les boutons et potentiomètres 3D de WAVE [117] ou les spectrogrammes 3D de OSE [32], les *widjets réactifs 3D* ont plusieurs avantages.

#### Organisation de l'interface

A l'inverse des interfaces qui répartissent les processus sur plusieurs composants graphiques, les *widjets réactifs 3D* se rapprochent des interfaces de type "Patch". L'agencement des widgets va en effet permettre de représenter l'organisation des processus sonores, c'est à dire les connexions et les flux de signal audio et de messages de contrôle. L'agencement spatial des représentations graphiques des processus va permettre de les mémoriser de manière efficace [36] [37].

#### Cohérence/Identification

D'un point de vue perceptif, l'utilisation d' "objets visuels" [77] associés aux "objets auditifs" [26] permettent de faciliter la séparation et l'identification de ces derniers comme l'ont montré Rahne et al. [104]. Si l'identification est améliorée, on peut supposer que l'interaction musicale en devient plus efficace car le musicien trouve plus rapidement les processus et paramètres qu'il souhaite modifier.

#### Degré d'indirection

Le degré d'indirection est un concept défini par Beaudoin-Lafon [7] afin d'évaluer les interfaces graphiques. Il représente la distance entre les objets d'intérêt, c'est à dire le contenu sur lequel porte les actions, et les composants manipulés pour effectuer ces actions sur le contenu. Par exemple, dans beaucoup de logiciels, la modification d'un élément passe par le parcours d'une série de menus, de fenêtres graphiques et de boutons.

Dans notre cas, les objets d'intérêt sont les représentations visuelles des processus sonores. Sur la plupart des instruments multi-processus ces représentations sont dispersées. Sur la figure 3.3, on peut voir un processus divisé en 3. Les paramètres de haut-niveau (séquence) sont affichés sur un composant de type linéaire. Les paramètres de bas-niveau (paramètres de synthèse) sont représentés par plusieurs potentiomètres. Finalement, la visualisation du résultat sonore est placée tout en haut de l'interface.

Le degré d'indirection est dans ce cas très important. En effet, les actions sont effectuées sur plusieurs éléments graphiques éloignés les uns des autres et surtout éloignés de la visualisation des processus. Ceci entraîne un partage de l'attention du musicien, et donc potentiellement une diminution de l'efficacité de l'instrument.

Les *widjets réactifs 3D* au contraire, combinent pour chaque processus tous les paramètres manipulables avec la visualisation de ce processus. Les manipulations s'effectuent directement sur les objets d'intérêt. Ainsi le degré d'indirection est réduit et l'efficacité de l'instrument augmentée.



FIG. 3.3 – Séparation des paramètres sur le logiciel LMMS

### Combinaison des paramètres

Finalement, tous les paramètres étant regroupés sur un même objet graphique, la visualisation des valeurs de plusieurs paramètres devient presque simultanée. Cette combinaison de paramètres est notamment exploitée dans le domaine de la visualisation d'informations car elle facilite la compréhension de grands ensembles de données multidimensionnels. Par exemple, Healey [64] utilise plusieurs paramètres graphiques (orientation, densité, couleur) de "coups de pinceaux" afin de représenter différents paramètres météorologiques simultanément.



FIG. 3.4 – Combinaison de paramètres graphiques : Visualisation picturale par Healey

Cette visualisation combinée peut permettre au musicien de facilement prévoir les changements à effectuer par rapport au résultat musical souhaité à un instant donné. Ces modifications peuvent d'ailleurs s'effectuer à une fréquence plus élevée, la distance (graphique) entre les paramètres étant réduite. Cependant, cela implique que les techniques d'interaction utilisées prennent en compte cette configuration, comme nous le verrons dans le chapitre 4.

## 3.2 Combinaison de mappings audiovisuels sur les *wid-gets réactifs 3D*

Les *widgets réactifs 3D* nous permettent de visualiser et de manipuler les paramètres de processus sonores par l'intermédiaire de paramètres graphiques. Il apparaît cependant essentiel de s'interroger sur le choix de ces paramètres sonores et graphiques. Peut-on utiliser n'importe lesquels ? Cela a-t-il une incidence sur l'efficacité des *widgets réactifs 3D* pour l'interaction avec les processus sonores ?

De même, une fois ces paramètres choisis, comment les associe-t-on ? Quel paramètre graphique est le plus pertinent pour représenter un paramètre sonore ?

Finalement, comment combiner plusieurs mappings audiovisuels sur un widget ? Cela perturbe-t-il la compréhension des différents paramètres qui les constituent ? Cela affecte-t-il l'efficacité des *widgets réactifs 3D* ?

Cette section apporte quelques éléments de réponse.

### 3.2.1 Paramètres Sonores

#### Quels paramètres sonores ?

- Différents types de paramètres sonores peuvent être visualisés et manipulés :
- Paramètres psychoacoustiques : hauteur, intensité, dimensions du timbre
  - Paramètres du signal : enveloppe, spectre
  - Paramètres de synthèse des différentes techniques : nombre, fréquences et amplitudes des sinusoïdes de la synthèse additive, et ainsi de suite.
  - Paramètres des effets audionumériques

Parmi ces possibilités, les paramètres perceptifs présentent l'avantage d'être compréhensibles, par rapport notamment à certains paramètres de synthèse tels que ceux de la synthèse granulaire par exemple. De plus ces paramètres perceptifs peuvent non seulement être extraits de n'importe quelle source sonore, mais ils peuvent également correspondre à certains paramètres d'effets audionumériques. Par exemple, le pitchshifting permet de modifier la hauteur d'un son, l'amplification permet de modifier le volume. Finalement, certaines techniques de synthèse, telles que la synthèse additive structurée [43], s'appuient sur ces paramètres perceptifs.

#### Perception auditive

Afin d'utiliser ces paramètres perceptifs, il faut connaître leur caractéristiques, et donc le fonctionnement du système auditif humain.

A un niveau physique, le son est porté par des variations de pression d'un matériau (liquide, solide, gazeux). On parle d'ondes sonores. Ces ondes, en arrivant à l'oreille, font vibrer le tympan. Ces vibrations sont ensuite transmises à la cochlée. Situé dans l'oreille interne, cet organe est un tube rempli de liquide, enroulé en spirale et parcouru par ce que l'on appelle la membrane basilaire. Comme l'ont montré les expériences

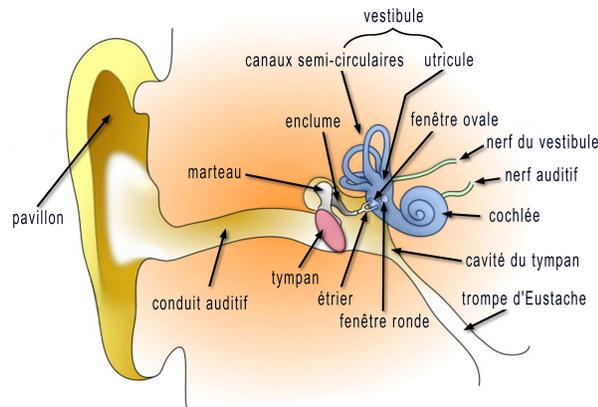


FIG. 3.5 – Principaux composants de l'oreille humaine.

de Békésy et al. [124], cette membrane se comporte comme un ensemble de filtres auditifs. En effet, son épaisseur et sa tension varient de sa base à son extrémité (apex). Pour une fréquence de vibration donnée, une partie de la membrane se déforme avec plus d'amplitude que les autres. Ainsi les fréquences graves affecteront plutôt la base de la membrane, tandis que les fréquences aiguës affecteront l'apex. Ces déformations sont finalement détectées par l'organe de Corti, composé de milliers de cellules sensorielles ciliées, et transformés en impulsions nerveuses transmises au cerveau par le nerf auditif. Nous pouvons ainsi percevoir les sons entre approximativement 20Hz et 20000Hz. Ces chiffres varient suivant les personnes, et suivant les dégradations de l'audition. Le temps d'intégration de l'oreille varie de 50 à 100 ms suivant l'intensité et nous pouvons détecter des fluctuations d'amplitude jusqu'à 1000 fois par seconde [5].

### Paramètres psychoacoustiques

Dans cette section, nous décrivons les différents paramètres perceptifs que nous allons représenter et manipuler.

La hauteur et le volume suivent la loi de Webner-Fechner, qui définit une relation logarithmique entre le stimulus et la perception pour la plupart des organes sensoriels.

#### – Volume :

Le volume est le corrélat perceptif de l'intensité d'un son. L'intensité est un paramètre physique proportionnel au carré de l'amplitude du signal sonore. Le volume correspond à la perception (logarithmique) de cette intensité et permet de placer les sons sur une échelle qui va d'un son faible à un son fort. Il est donné

en décibels (dB), et est calculé par rapport à l'amplitude du signal par :

$$v = 20 \log_{10} \left( \frac{a}{A_0} \right)$$

Si on définit que l'amplitude maximale doit correspondre à un volume de 120dB, l'amplitude de référence est alors égale à  $A_0 = 10^{-6}$ .

Le volume peut également être calculé à partir de la mesure de l'intensité par rapport à une intensité de référence, où de la mesure d'une pression sonore par rapport à une pression de référence. Ainsi, on obtient le volume en dB SPL (Sound Pressure Level) par :

$$v = 20 \log_{10} \left( \frac{p}{p_{ref}} \right) = 10 \log_{10} \left( \frac{I}{I_{ref}} \right)$$

où  $p_{ref}$ , la pression sonore de référence, est habituellement défini à  $20 \mu\text{Pa}$  dans l'air (et  $1 \mu\text{Pa}$  dans l'eau), et  $I_{ref}$  est défini comme  $10^{-12} \text{W.m}^{-2}$ . Ces valeurs prennent pour référence le seuil d'audibilité d'une fréquence de 1000Hz.

Le volume perçu peut finalement s'exprimer par le paramètre subjectif de sonie, dont l'unité est le sone [112]. Un sone correspond à 40 phones, valeur définie comme l'intensité d'un son pur de fréquence 1kHz à 40dB SPL. La sonie permet de prendre en compte les différences de perception du volume suivant les fréquences. Celles-ci sont liées à plusieurs paramètres, notamment la résonance du conduit auditif. Ainsi le seuil d'audibilité et la perception de l'intensité varie en fonction des fréquences, comme présenté sur la figure 3.6.

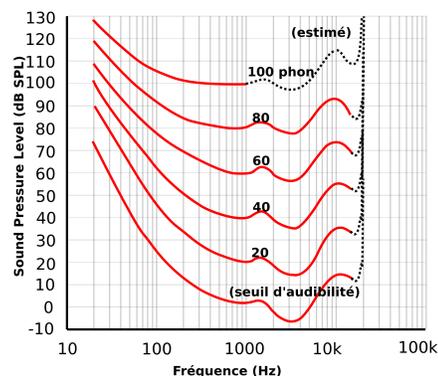


FIG. 3.6 – Lignes isosoniques (en phones) et seuil d'audibilité.

Les modulations du volume sont des composantes essentielles de la perception musicale. Ainsi une vibration sinusoïdale (d'une fréquence de 10Hz) du volume est appelé trémolo. Le début (onset) des sons et la fin (offset) des sons permettent

d'identifier les objets sonores.

– Hauteur :

La hauteur est le paramètre du son qui nous permet de le classer de grave à aigu. Dans le cas d'un son pur la hauteur perçue correspond à la perception de la fréquence de l'onde. Les sons complexes, dits harmoniques, sont composés de plusieurs fréquences multiples d'une fréquence fondamentale, que l'on appelle harmoniques. La hauteur perçue correspond alors à cette fondamentale, ou à la combinaison des autres harmoniques. L'être humain est capable de distinguer plus de 600 hauteurs différentes, et des mélodies situées entre 30Hz et 5000Hz [4]. Le seuil temporel de reconnaissance de la hauteur est définie à 10ms en moyenne, un son plus bref n'a pas de hauteur.

Dans la musique occidentale, les mélodies sont constituées de notes, 12 notes donnant une octave. Puisque notre perception des fréquences est logarithmique, monter d'une octave revient à multiplier la fréquence par deux. Dans la norme MIDI, on obtient ainsi la fréquence d'une note par :

$$f = 2^{(n-69)/12} * 440Hz$$

où n est la note, avec 69 correspondant au LA de la sixième octave, ceci dans le cas où ce LA est défini à 440Hz.

La hauteur peut également être représentée par les échelles de Mel et de Bark. L'échelle de Mel [123] est définie afin d'obtenir une variation linéaire de la hauteur. Elle prend pour référence un son pur de fréquence 1000Hz à 40dB, qui correspond à 1000mel, et est obtenue en demandant à des sujets d'évaluer la distance entre des fréquences.

Une autre échelle, encore une fois proche de la perception, est celle de Bark. Elle s'appuie sur la notion de bandes critiques définis par Zwicker [139]. Ces bandes expliquent entre autre les phénomènes de masquage fréquentiel. En effet, un son faible ne sera pas perçu s'il se trouve dans la même bande critique qu'un son plus fort, car ce son provoque un masquage (triangulaire dans l'échelle de Bark). 24 bandes critiques suffisent à couvrir le spectre de manière linéaire et analogue à la perception humaine. Les fréquences sont converties en Bark par la formule suivante :

$$z = \frac{26.81}{1 + \frac{1960}{f}} - 0.53$$

– Timbre :

Le timbre est souvent défini comme le paramètre qui permet de différencier deux sons de même volume et de même hauteur. Il dépend de l'organisation du spectre, c'est à dire des différentes fréquences qui composent le son. Nous

avons une bonne perception des variations de spectre, donc des changements d'amplitudes de chaque fréquence d'un spectre, même si ces changements sont faibles. Cette propriété est appelée "analyse de profile" [58] et nous permet d'identifier rapidement les modifications de timbre d'un son.

On peut donner deux exemples de descripteurs perceptifs du timbre tirés de [75] : La brillance correspond à la perception du centroïde spectral. Le centroïde  $c$  est le barycentre du spectre et se calcule par la formule :

$$centr = \frac{\sum_{k=1}^N k a_k}{\sum_{k=1}^N a_k}$$

La caractèrè plus ou moins bruité (donc plus ou mois pur) d'un son peut être calculé grâce à l'irrégularité spectrale [70].

$$irreg = \frac{\sum_{k=1}^N (a_k - a_{k+1})^2}{\sum_{k=1}^N a_k^2}$$

Krimphoff utilise également l'attaque du son comme descripteur du timbre.

La forme du spectre peut être globalement défini en utilisant les MFCC (Mel Frequency Cepstral Coefficients). Ce sont les coefficients du cepstrum (transformée de Fourier du spectre du son) sur l'échelle de Mel.

– Localisation :

La position spatiale d'un son est perçue grâce aux différences de temps d'arrivés (ITD) et de volume (ILD) d'un son entre les deux oreilles. En effet, un son provenant par exemple de la droite sera perçu avant et plus fort par l'oreille droite que par l'oreille gauche. Pour les basses fréquences (en dessous de 800Hz), la différence de temps est utilisée pour la localisation, tandis que pour les hautes fréquences (au dessus de 1600Hz), seule la différence d'amplitude est utilisée. Le système auditif humain a une précision d'un degré.

Suivant certaines modélisations de la perception humaine, ces paramètres sont organisés de manière hiérarchique [90]. Ils permettent de séparer les sources sonores que nous percevons simultanément et de suivre les objets auditifs, ce que Bregman appelle l'analyse de scène auditive [26]. Des sons sont identifiés comme appartenant à une même source s'ils commencent en même temps (onset) et si leurs contenu harmonique correspond. Le système auditif cherche ensuite des similarités entre les sons afin de suivre ces objets auditifs, notamment au niveau de leur spectre, de leur fréquence fondamentale et de leur localisation.

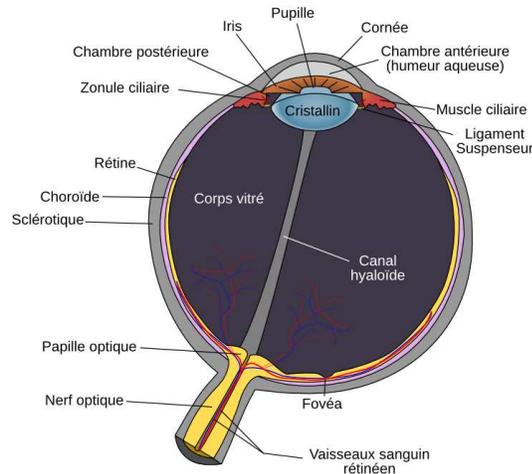


FIG. 3.7 – Principaux composants de l'œil humain.

### 3.2.2 Paramètres Graphiques

#### Perception visuelle

La lumière arrive dans l'œil par la pupille qui se situe au milieu de l'iris. La pupille se dilate et se contracte en fonction de la luminosité ambiante. Les rayons lumineux sont décomposés par les cônes et les bâtonnets situés sur la rétine. Les cônes sont utilisés pour la vision diurne. Il en existe 3 types, qui réagissent chacun à une bande de longueurs d'ondes. Les cônes cyanolabe réagissent autour de la longueur d'onde 424nm, les cônes chlorolabe autour de 530nm et les cônes érytholabe autour de 560nm. Les bâtonnets sont utilisés pour la vision nocturne, ils détectent uniquement la luminance. Ces signaux sont ensuite convertis en trois nouveaux types avant d'être envoyés au nerf optique sous forme d'influx nerveux.

#### Paramètres perceptifs visuels

De la même manière que la perception auditive, la perception visuelle est organisée de manière hiérarchique. Selon la théorie de l'intégration des propriétés (Feature Integration Theory) de Treisman [93], plusieurs paramètres sont traités en parallèle. Ainsi, au niveau le plus bas, les tailles, orientation, textures, couleurs et mouvements sont détectés. A un niveau supérieur, les motifs 3D sont détectés, ainsi que les contours, les régions et les silhouettes. Finalement, se forme la perception de l'espace visuel 3D et les formes complexes 3D à partir des primitives.

Beaucoup de ces propriétés sont traitées de manière pré-attentionnelle. Elle sont ainsi détectées très rapidement (moins de 200 à 250ms) et précisément par le

système visuel. Les tâches utilisées pour évaluer ces propriétés sont la recherche d'un objet parmi un ensemble, la recherche de frontières, et le comptage d'objets. Ces propriétés pré-attentionnelles sont recensées par Healey et al. dans [65]. On trouve ainsi : orientation des lignes/blobs, fermeture, taille, courbure, densité/contraste, nombre/estimation, couleur/teinte, intensité, intersection, terminaisons, indices de profondeur/profondeur stéréoscopique, clignotement (jusqu'à 50Hz), direction et vitesse du mouvement, direction d'éclairage, orientation 3D. Les combinaisons de propriétés ne sont pas pré-attentionnelles bien que certaines études telles que celle de McLeod et al. [89] montrent que par exemple le mouvement et la forme sont détectés en parallèle. A un plus haut niveau, les objets visuels sont finalement identifiés et stockés dans la mémoire à court terme. Selon Luck et al. [82], les objets sont stockés de manière intégrale (les propriétés ne sont pas séparées). Il est ainsi possible d'utiliser jusqu'à quatre propriétés par objet sans perte de performances.

Les propriétés de bas niveau, particulièrement les propriétés pré-attentionnelles peuvent être utilisés pour représenter de manière efficace de grands ensembles de données. Elle sont ainsi employées dans le domaine de la visualisation d'informations qui s'appuie sur les recherches concernant la perception visuelle, comme le décrit Ware [130] Dans cette même optique, elles sont combinées, par exemple sur les "pexels" de Healey [63] ou les formes 3D d'Ebert et al. [45], afin de représenter plusieurs dimensions simultanément. Nous allons maintenant détailler les caractéristiques de certaines de ces propriétés, et les spécificités liées à leur utilisation.

– Couleur :

Il faut distinguer l'utilisation de la couleur pour représenter plusieurs valeurs discrètes, auquel cas on parle de codage nominal, de son utilisation sous forme d'échelles de valeurs continues (codage ordinal). Pour un codage nominal, il est important de restreindre le nombre de couleurs utilisées, car au-delà d'une dizaine de valeurs la mémorisation devient problématique. De plus, alors que l'être humain ne peut détecter que quelques dizaines de niveaux de gris différents, il est possible d'arriver à des centaines de couleurs séparables en utilisant des échelles faisant varier plusieurs dimensions et basées sur la perception. Ainsi l'encodage des couleurs doit se faire de préférence sur des espaces colorimétriques proches de la perception telles que le CIELUV défini par la Commission internationale de l'éclairage en 1976, ou le CIELab. Ce dernier espace spécifie les coordonnées de luminance  $L^*$ , et de chrominance par  $a^*$  et  $b^*$ .  $a^*$  définit la position entre les couleurs rouge/magenta et vert, et  $b^*$  entre les couleurs jaune et bleu.

Ware introduit ainsi une échelle de couleurs qui forme une spirale autour de l'axe de luminance [129]. Healey [63] propose de choisir ces couleurs pour la visualisation en contrôlant la distance de couleur, la séparation linéaire et la catégorie de couleur. Zhang et al. [138] définissent plusieurs échelles, présentées sur la figure 3.8. Leur étude révèle que les échelles s'appuyant sur la luminance  $L^*$  et des variations de chrominance, et les échelles divergentes sont significativement préférées pour la perception de détails.

Il est également possible d'utiliser des échelles à deux dimensions, qui per-

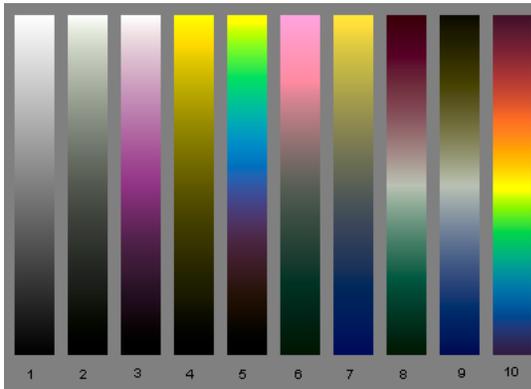


FIG. 3.8 – Échelles testées par Zhang et al. dans [138]. Du niveau de gris avec un espace de couleur RGB (1), puis avec luminance du CIELab (2), jusqu'à une échelle spectralement divergente, en passant par des variations de luminance  $L^*$  alliée à des variations de chrominance (5).

mettent de visualiser deux dimensions à partir d'une couleur.

– Texture :

Comme décrit par Ware [131] les textures possèdent plusieurs dimensions, notamment la densité, la direction, la régularité. On peut également ajouter le contraste, la taille, la rugosité (pour un rendu de type bump-mapping).

– Position dans l'espace :

Selon [37], l'agencement spatial améliore la mémorisation des données. Certaines études montrent un bénéfice des représentations 3D qui réduirait le nombre d'erreurs de mémorisation et de perception structurelle, mais d'autres études [36] ne montrent aucune différence avec les représentations 2D.

La position dans l'espace est perçue grâce à la vision stéréoscopique (vue différente avec chaque oeil), et grâce à des indices tels que les ombres, les mouvements et les tailles relatifs. En effet, un objet plus gros et qui se déplace plus vite est perçu comme étant plus proche.

– Forme 3D :

La forme 3D est détectée en utilisant les informations de courbure des ombres, les déformations de textures, et la vision stéréoscopique. Elle peut servir à représenter plusieurs dimensions, soit par la taille des éléments d'une forme composite, soit par différentes formes générées. Par exemple, Ebert et al. [45] présente trois types de formes procédurales. Les superquadriques, représentées sur la figure 3.9 permettent d'afficher une ou deux dimensions, les formes frac-

tales de quatre à six dimensions, et les surfaces implicites de huit à 20 dimensions.

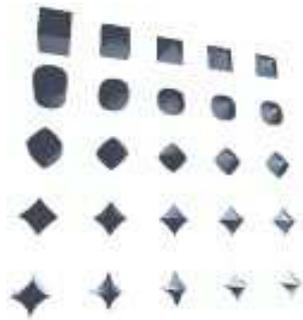


FIG. 3.9 – Exemples de formes superquadriques [45]

### 3.2.3 Travaux Existants

Parmi les travaux existants sur les mappings audiovisuels [79], on trouve des recherches artistiques, des logiciels de sonification et de visualisation, et des tests utilisateurs.

#### Artistiques

Dans le cadre artistique, il faut citer les travaux de John Whitney [134], un des pionniers de l'animation numérique, et qui a beaucoup exploré les relations entre couleurs/formes et tonalités. Michel Chion [33] a fourni un travail considérable sur les relations entre le son et l'image notamment cinématographique. Il définit ainsi certains principes ou associations audiovisuels tels que l'aimantation spatiale, illusion qui nous fait percevoir un son comme provenant de la source représentée, ou encore la synchrèse, phénomène correspondant à une impression de causalité entre des événements audiovisuels lorsque ceux-ci se produisent simultanément.

Un exemple très intéressant de travail sur les représentations 3D de la musique est le clip vidéo du titre Gantz Graf<sup>1</sup> du groupe Autechre, réalisé par Alex Rutterford, et qui utilise de nombreux mappings audiovisuels. Dans ce clip vidéo, une forme 3D composée évolue en réagissant aux différents paramètres et événements musicaux de la piste. Toutes les modifications des formes ont cependant été définies et synchronisées manuellement, et non automatiquement ou en temps réel.

#### Logiciels

Deux grandes catégories de logiciels explorent des mappings audiovisuels autre que les représentations classiques telles que les enveloppes, spectre, spectrogramme. Les

<sup>1</sup>[http://en.wikipedia.org/wiki/Gantz\\_Graf](http://en.wikipedia.org/wiki/Gantz_Graf)

logiciels de sonification interactive [108] [2] et de dessin musical [105] [80] utilisent les formes et les textures comme des partitions à parcourir. D'autres tels que [91], inspiré par le travail de Chion [33], associent les paramètres sonores aux mouvements des objets graphiques.

Parallèlement on trouve des logiciels de visualisation tels que [46] pour la visualisation du son produit par un instrumentiste ou [66] pour la visualisation, cette fois en 3D, de données de performances musicales.

Tous ces logiciels s'appuient principalement sur des mappings définis de manière arbitraire et non sur des tests utilisateurs. Certains n'utilisent pas non plus les paramètres perceptifs.

### Tests utilisateurs

De nombreux tests utilisateurs ont été menés afin de déterminer les préférences des utilisateurs en terme de mappings audiovisuels. Certains, menés par exemple par Caivano [30], s'appuient sur des analogies entre les dimensions physiques (dans le cas de Caivano, entre la couleur et le son). Nous nous concentrons plutôt sur les tests s'appuyant sur les paramètres perceptifs, pour les raisons expliquées dans les sous-sections précédentes. Les préférences significatives trouvées par les différents auteurs sont représentées sur la figure 3.10.

Walker [126] étudie les préférences de sujets musiciens et non-musiciens avec des séquences de signes 2D dont les paramètres sont la taille, la position verticale, le motif et la longueur horizontale. Les paramètres sonores sont le volume, la hauteur, le timbre (forme d'onde simple, son complexe et bruit). Il trouve des préférences significatives mais seulement pour les musiciens.

Les travaux de Lipscomb et Kim [81] reprennent les mêmes procédures et paramètres, en remplaçant cependant le motif visuel par la forme et en ajoutant la teinte. Les stimuli sont des séquences de trois étapes de variation des paramètres. Les résultats sont similaires à ceux de Walter bien que la teinte soit également associée à la hauteur et au volume. Lipscomb ne trouve pas de différences entre les musiciens et non-musiciens.

Finalement, Giannakis et Smith [56, 55, 54] conduisent une série de tests. Les stimuli visuels sont des dimensions perceptives de la couleur et de la texture. Giannakis s'appuie ainsi sur les recherches en visualisation d'informations. A l'aide d'une palette (pour le test sur la couleur), les sujets doivent créer des séquences de six couleurs correspondant à des séquences de six sons. Ces suites de sons comprennent des variations linéaires ou aléatoires de paramètres sonores. Cette fois la hauteur et le volume sont respectivement associés à la luminance et saturation de la couleur et les différentes dimensions du timbre sont associées aux différentes dimensions de la texture. La teinte est également associée à la hauteur mais de manière non significative.

On observe donc plusieurs points communs entre ces travaux. En particulier, l'association volume/taille et la correspondance entre le timbre et la texture reviennent. Mais il apparaît également que selon les études différents mappings sont préférés. Ainsi, la hauteur peut être associée à la luminance comme à la position verticale, et le

Auteur	Paramètre Auditif	Paramètre Visuel
Walker[126]	volume	taille
	hauteur	position verticale
	timbre	motif
	durée	longueur horizontale
Lipscomb et Kim [81]	volume	taille
	volume	teinte
	hauteur	position verticale
	hauteur	teinte
	timbre	forme
Giannakis et Smith [56]	volume	saturation
	hauteur	luminance
	dissonance	répétitivité
	finesse	rugosité
	compacité	granularité

FIG. 3.10 – Préférences de mappings résultants des travaux existants

volume à la saturation comme à la taille.

Cependant ces tests s'appuient uniquement sur des représentations graphiques statiques en deux dimensions. Ils n'évaluent pas non plus les performances des mappings, par exemple pour des tâches d'identification de sources sonores, mais uniquement les préférences des utilisateurs.

### 3.2.4 Tests utilisateurs : Préférences, Performances et Combinaisons

#### Hypothèses

Nous proposons trois expérimentations pour évaluer les mappings audiovisuels dynamiques sur des objets 3D.

La première consiste à vérifier les préférences des sujets. Notre hypothèse est que nous allons trouver des résultats similaires aux travaux existants.

La deuxième expérimentation évalue les performances de plusieurs associations, sélectionnées d'après les résultats de la première, dans une tâche d'identification d'objet 3D audiovisuel. Notre hypothèse est que certains mappings sont plus efficaces que d'autres pour cette tâche.

La dernière expérimentation concerne les combinaisons de mappings audiovisuels, utilisés notamment pour les *widgets réactifs 3D*. Notre hypothèse est que la combinaison de mappings audiovisuels améliore les performances pour la même tâche d'identification par rapport à des mappings uniques.

### Stimuli visuels et sonores

Nous évaluons quatre paramètres psychoacoustiques pour les trois expérimentations, dix paramètres visuels pour la première, cinq pour la deuxième et quatre pour la troisième.

Sept séquences sonores d'une durée de sept secondes sont générées pour chaque paramètre sonore. Pour chaque séquence, une sinusoïde est synthétisée par table d'onde et modifiée à l'aide d'effets simples afin de produire les variations des paramètres sonores. Sept autres séquences sont générées pour la troisième expérimentation en combinant des variations des quatre paramètres sonores. Ces paramètres sont ensuite analysés à l'aide du logiciel Pure-Data<sup>2</sup> afin d'extraire les valeurs qui sont visualisées par les paramètres graphiques.

- Le premier paramètre est la hauteur. Les variations sont obtenues par modification de la vitesse de lecture de la sinusoïde. Elles sont choisies aléatoirement pour chaque séquence générée parmi les notes {60, 63, 67, 68, 72} qui correspondent aux fréquences {261.6Hz, 311.1Hz, 391.9Hz, 415.3Hz, 523.2Hz}. Les variations sont faibles pour éviter des modifications du volume perçu provoqué par les fréquences plus élevées. La hauteur est extraite grâce à l'external Fiddle~ développé par Puckette et al. [102].
- Les variations de volume, discrètes et continues, sont obtenues par un simple effet d'amplification (multiplication du signal). Elles sont générées aléatoirement entre 0dB et 30dB et analysés par l'objet env~.
- Les variations du centroïde spectral sont obtenues par une saturation de la sinusoïde et l'application d'un filtre passe-bas. Des variations aléatoires discrètes et continues de la fréquence de coupure sont analysées par l'external sc~, de la bibliothèque flib, comme des changements de la brillance entre 948Hz et 998Hz.
- Les variations aléatoires discrètes et continues de l'irrégularité sont obtenues en mixant la sinusoïde avec du bruit blanc filtré avec un filtre passe-bande avec une fréquence de coupure égale à la fréquence de la sinusoïde. Elle est analysée par l'external irreg~, selon [70], comme s'étalant entre 3.6 et 170.

Le problème pour générer ces paramètres sonores est que la variation d'un paramètre peut entraîner celle d'un autre.

Lors de la lecture des séquences, les valeurs des paramètres sonores, extraites en utilisant le logiciel Pure-Data, sont envoyées grâce à des messages du protocole OpenSoundControl et modifient les valeurs des paramètres graphiques. Les paramètres graphiques sont appliqués en temps réel à des cubes positionnés côte à côte dans une scène 3D, au dessus d'un plan représentant le sol. Le rendu de cette scène est effectué par une application utilisant la bibliothèque de graphes de scène OpenSG<sup>3</sup>, s'appuyant elle-même sur OpenGL. Ces cubes projettent des ombres sur le sol, comme l'illustre la figure 3.11. Pour les paramètres de texture, une texture de brique est appliquée sur les cubes.

<sup>2</sup><http://puredata.info>

<sup>3</sup><http://opensg.vrsource.org>

- Le paramètre de distance correspond à l'éloignement du cube sur l'axe Z (profondeur). Il varie des positions -1 à -3, la caméra se trouvant en 0. La visualisation de ce paramètre est facilitée par la présence des ombres.
- Le paramètre d'orientation est une rotation autour de l'axe Z entre les valeurs 0rad et 1rad.
- Le paramètre de taille correspond à l'échelle du cube (sur les trois axes). Ce paramètre varie entre 0.4 et 0.9.
- Les variations de rugosité de la texture sont obtenues en modifiant le paramètre d'échelle du rendu en bump-mapping. Cette échelle s'étale de 0 (texture plate) à 10 (texture très creusée).
- Le paramètre de granularité de la texture correspond à la taille de cette texture. Les briques varient de leurs tailles initiales à huit fois celles-ci.
- Le paramètre de vitesse correspond à un déplacement circulaire autour de l'axe Z. Il varie entre 0 (cube immobile) et  $\pi$  radians par seconde.
- Seul le paramètre de luminance est testé et non la teinte, ni la saturation. Ce paramètre correspond donc à une variation de la couleur de l'objet entre les valeurs RVB  $\{0, 0, 0\}$  et  $\{255, 255, 255\}$ .
- Le paramètre de brillance correspond à des variations de l'intensité de réflexion spéculaire de la surface des cubes. Il varie entre complètement mat et très brillant.
- Le paramètre de distorsion est une modification aléatoire de la position des sommets du cube, plus ou moins amplifié.
- Finalement, la transparence des cubes varie de 0 (opaque) à 0.8 (presque complètement transparent). Pour s'assurer de la perception correcte de la transparence, un fond bruité est ajouté à la scène.

Les positions verticales et horizontales des cubes n'ont pas été testées. Ces paramètres ont en effet déjà été étudiés pour des mappings audiovisuels 2D.

### Sujets et dispositif expérimental

Les 16 sujets, treize hommes et trois femmes, ont entre 22 et 54 ans. Ils sont tous des musiciens entraînés, ayant au moins pris des leçons pour leur instrument, et ont tous déjà utilisé des logiciels de musique. Aucun d'entre eux n'est un utilisateur régulier d'applications 3D, que ce soit des jeux ou des applications de modélisation.

Les sujets sont assis en face d'un écran d'ordinateur portable de 12,1", et sont équipés d'un casque Beyerdynamic DT-770. Ils complètent les expérimentations en utilisant un clavier d'ordinateur modifié sur lequel toutes les touches ont été enlevées sauf la barre d'espace et quatre touches. La barre d'espace permet de lancer une séquence et les quatre touches permettent pour les deuxième et troisième expérimentations de choisir un objet audiovisuel 3D.

Les trois expérimentations sont réalisées à quelques semaines d'intervalle. Les sujets effectuent tous la première, puis reviennent pour effectuer la deuxième, et ainsi de suite. Ceci nous a permis par exemple d'utiliser les résultats de la première expérimentation pour mettre en place la deuxième.

### Première expérimentation : Préférences

**Procédure** La première expérimentation se déroule en quatre étapes, une pour chaque paramètre sonore. Lors de chacune de ces étapes, une des sept séquences créées pour ce paramètre est lue, et dix objets audiovisuels 3D sont affichés, chacun avec un mapping différent, c'est à dire associant le paramètre variant dans la séquence sonore lue à un paramètre graphique différent. Ceci est illustré par la figure 3.11. Les sujets évaluent la pertinence de chaque mapping et donnent une note entre 1 et 5. On obtient ainsi une note pour les dix paramètres graphiques pour chaque paramètre sonore, donc 40 notes par sujet.

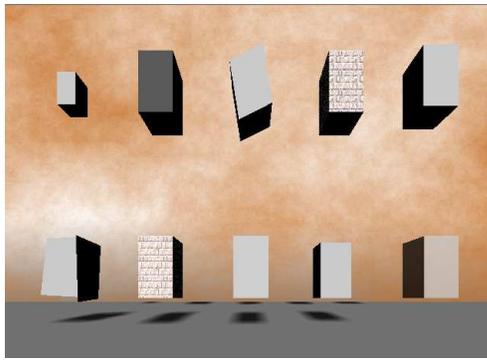


FIG. 3.11 – Capture d'écran de l'expérimentation sur les préférences.

**Résultats** Les résultats de la première expérimentation sont présentés sur les figures 3.12, 3.13 et 3.14. Des tests de Kruskal-Wallis révèlent un effet significatif des paramètres graphiques pour le Volume ( $\chi^2(9) = 57.81, p < 0.0001$ ), la hauteur ( $\chi^2(9) = 32.178, p = 0.0002$ ), le centroïde spectral ( $\chi^2(9) = 29.932, p = 0.0005$ ), mais pas pour l'irrégularité. Les différences significatives révélées par les tests de Wilcoxon pour chaque paire de mappings sont précisées sous chaque figure.

### Deuxième expérimentation : Performances

**Procédure** La deuxième expérimentation porte sur les performances des mappings possibles entre les quatre paramètres sonores, et cinq paramètres graphiques choisis pour chaque paramètre sonore suivant les résultats de la première expérimentation. Les paramètres sélectionnés sont :

- Volume : taille, distance, luminance, transparence et brillance
- Hauteur : distance, luminance, brillance, taille et orientation
- Centroïde spectral : luminance, brillance, taille, transparence et distorsion
- Irrégularité : Rugosité de la texture, luminance, distorsion, taille et transparence

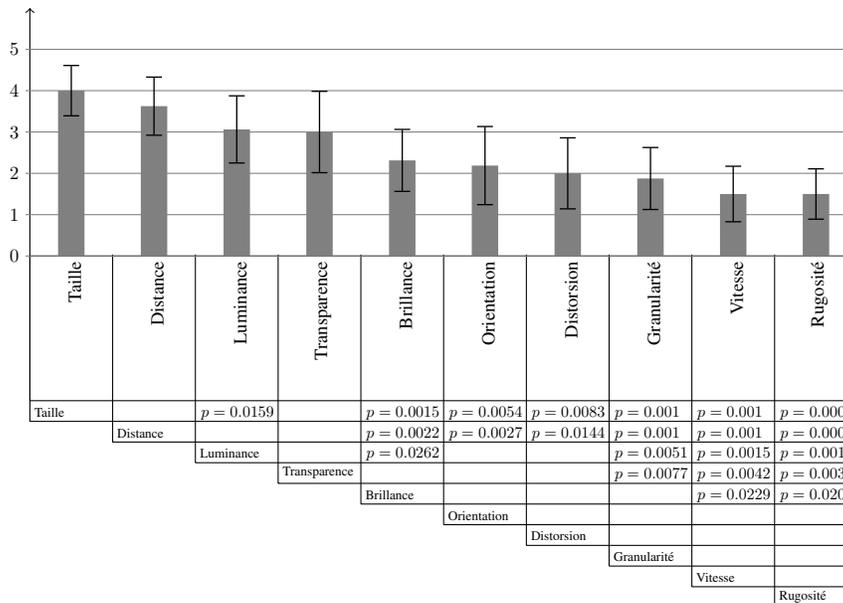


FIG. 3.12 – Première expérimentation : Moyennes des évaluations par les sujets pour le Volume (et déviations moyennes). Les différences significatives pour chaque paire de mappings sont également données.

Cette expérimentation se déroule donc en quatre étapes, une par paramètre sonore. Chaque étape contient cinq conditions, une par paramètre graphique, et chaque condition contient cinq essais. Comme on peut le voir sur la figure 3.15, cette fois quatre cubes sont affichés. Pour chaque essai une des sept séquences sonores, choisie aléatoirement, correspondant au paramètre sonore est lue. Seul un des cubes, défini aléatoirement, est associé à cette séquence, c'est à dire que le paramètre graphique de la condition suit les variations du paramètre sonore. Les paramètres des trois autres cubes varient selon des séquences sonores que le sujet n'entend pas. Pour chaque essai, le sujet déclenche la lecture de la séquence sonore avec la barre d'espace puis il doit identifier le plus vite possible lequel des quatre cubes est associé à la séquence sonore, en utilisant les quatre touches du clavier. Nous enregistrons le temps de réponse, et les erreurs. A la fin de chaque condition nous demandons également au sujet d'évaluer la difficulté d'identification pour ce mapping. Afin d'éviter un effet d'apprentissage, l'ordre des séquences et des conditions pour chaque paramètre sonore est choisi aléatoirement. Au début de chaque condition un essai d'entraînement est également ajouté.

**Résultats** Les temps de réponse, et les déviations moyennes, pour la deuxième expérimentation sont présentés sur la figure 3.16. Une analyse de variance (ANOVA) indique un effet significatif du choix du mapping pour le volume ( $F = 6.131 >$

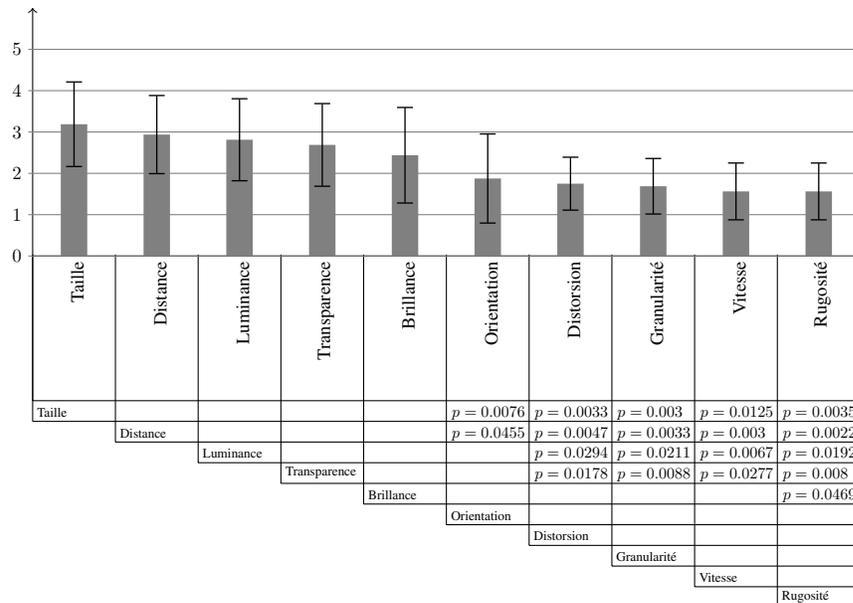


FIG. 3.13 – Première expérimentation : Moyennes des évaluations par les sujets pour la Hauteur (et déviations moyennes). Les différences significatives pour chaque paire de mappings sont également données.

$F(4, 60, 0.05) = 2.525$ ), la hauteur ( $F = 4.132 > F(4, 60, 0.05) = 2.525$ ), et l'irrégularité ( $F = 4.025 > F(4, 60, 0.05) = 2.525$ ), mais pas d'effet significatif pour le centroïde spectral. Un test de Student-Neumann-Keuls pour chaque pair de mappings révèlent les différences significatives suivantes :

- Volume : La taille, la distance, la transparence et la luminance donnent de meilleurs temps que la brillance
- Hauteur : La distance donne de meilleurs temps que l'orientation, et à la fois la distance et l'orientation donne de meilleurs performances que la brillance
- Irrégularité : La taille donne de meilleurs temps que la rugosité

L'ANOVA ne donne pas d'effet significatif pour les taux d'erreurs.

### Troisième expérimentation : Combinaisons

**Procédure** La procédure de la troisième expérimentation est sensiblement la même que celle de la deuxième. A nouveau le sujet doit identifier le cube associé à la séquence sonore jouée. Cette expérimentation se déroule en deux parties.

La première partie a pour but d'évaluer la perturbation d'un mapping par des variations aléatoires d'autre paramètres graphiques. Une séquence comportant des variations de volume est jouée. Dans la première condition, seul le mapping volume-taille est activé. Pour les conditions suivantes, à chaque fois des variations aléatoires (non

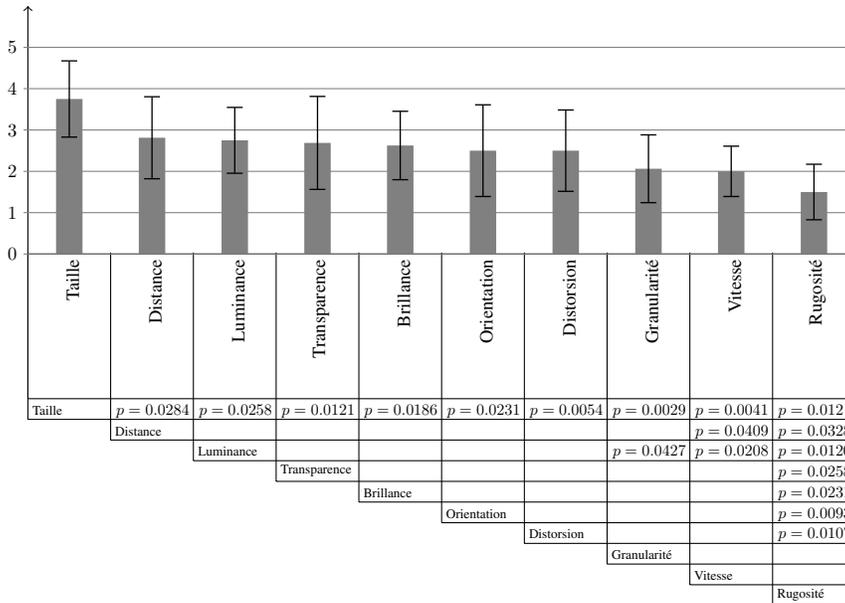


FIG. 3.14 – Première expérimentation : Moyennes des évaluations par les sujets pour le Centroïde spectral (et déviations moyennes). Les différences significatives pour chaque paire de mappings sont également données.

connectées aux paramètres sonores) d'un paramètre graphique sont ajoutées à tous les cubes.

La deuxième partie a pour objectif de tester si l'ajout de mappings pour une séquence sonore complexe améliore l'identification des objets audiovisuels 3D. Cette fois cependant la séquence est choisie parmi les séquences composées (variations des quatre paramètres sonores simultanément). Seul le mapping volume-taille est activé pour la première condition. Dans les suivantes, les mappings {hauteur/luminance, brillance/distorsion, irrégularité/transparence} sont respectivement ajoutés, comme on peut le voir sur la figure 3.17. Encore une fois, seul un des cubes est connecté à la séquence sonore, les paramètres graphiques des autres cubes varient suivant des séquences que le sujet n'entend pas.

Cette expérimentation est ensuite conduite une deuxième fois, en remplaçant le paramètre de luminance par la distance, afin de tester les interférences entre les variations de distance et de taille.

**Résultats** Les temps de réponse, et les déviations moyennes, pour la première partie de la troisième expérimentation, qui évalue les perturbations des mappings audiovisuels, sont donnés sur la figure 3.18. Une analyse de variance indique un effet significatif du nombre de paramètres graphiques perturbant le mapping, aussi bien lors de l'utilisation de la distance ( $F = 9.498 > F(3, 45, 0.05) = 2.812$ ) que lors de l'utilisation de la luminance ( $F = 10.094 > F(3, 45, 0.05) = 2.812$ ). Des tests

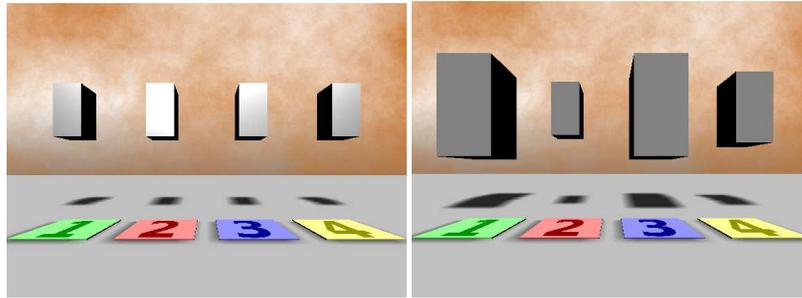


FIG. 3.15 – Capture d’écran de l’expérimentation sur les performances avec les paramètres de luminance et de taille.

de Student-Neumann-Keuls révèlent une différence entre la première condition et les trois autres lorsque la distance est utilisée, et des différences entre les trois premières conditions et la dernière dans le cas de la luminance.

Les temps de réponse, et les déviations moyennes, pour la deuxième partie de la troisième expérimentation sont donnés sur la figure 3.19. Une analyse de variance ne révèle aucun effet significatif de l’ajout des mappings sur les temps de réponse, aussi bien lors de l’utilisation de la distance que de la luminance.

### 3.2.5 Analyse et Discussion

#### Résultats et Limites des tests effectués

Les résultats de la première expérimentation indiquent des préférences similaires à celles retrouvées dans les différents tests utilisateurs effectués par Walker et Giannakis sur des représentations 2D. En particulier les sujets ont associé le volume à la taille des cubes. Ils ont également associé la distance à la hauteur en commentant pour certains que ce déplacement est assimilé à un potentiomètre linéaire comme le serait un déplacement vertical. De même l’orientation peut être assimilée selon certains sujets à un potentiomètre linéaire. Comme dans les tests précédents, les résultats et les commentaires montrent une tendance des sujets à associer l’aspect (texture/forme) au timbre, perçu comme l’identité du son. Des préférences liées à des analogies avec le monde réel et les propriétés physiques, ou logiques sont également présentes. Ainsi les sujets associent la distance et la transparence au volume. Cependant, des problèmes d’échelles apparaissent dans les commentaires des sujets, notamment sur ces analogies. Par exemple, les sujets auraient préféré que l’échelle de la taille soit inversée lorsqu’elle représente la hauteur, afin qu’une hauteur plus élevée soit représentée par un objet plus petit.

Une analyse comparée des résultats de la première et de la deuxième expérience nous montre qu’il n’y a pas de corrélation entre les préférences et les performances.

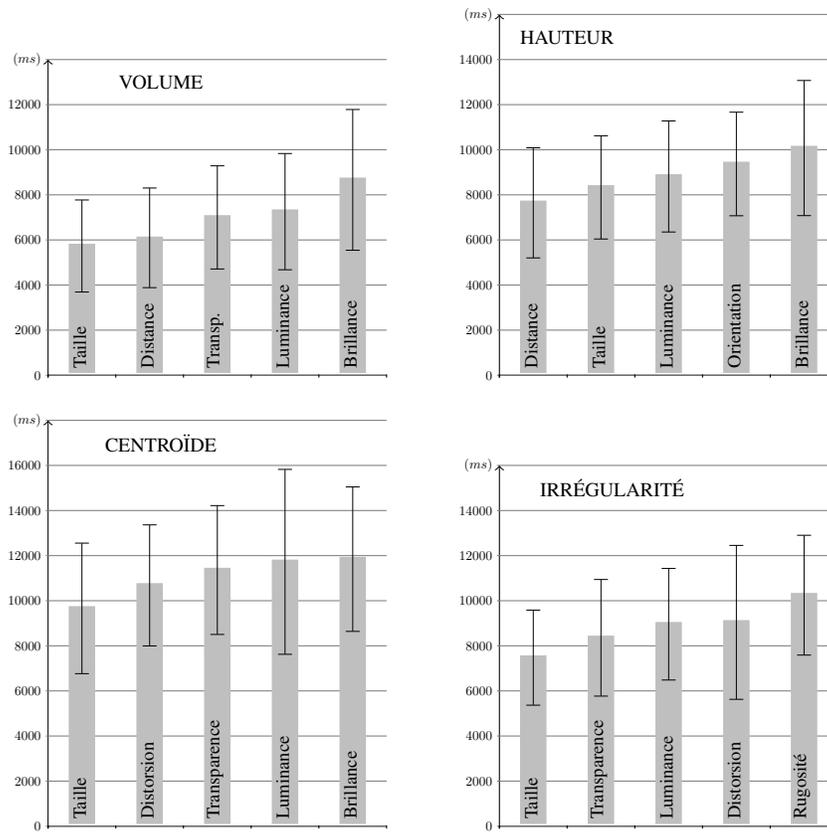


FIG. 3.16 – Deuxième expérimentation : Temps de réponse moyens (en ms) et déviations moyennes pour le volume, la hauteur, le centroïde spectral et l'irrégularité

Les sujets rapportent encore une fois des problèmes d'échelles, notamment sur la rugosité, la transparence et la brillance. Ces problèmes peuvent expliquer les quelques différences significatives trouvées entre les mappings. En effet, aucune différence n'apparaît en dehors de celles impliquant des paramètres dont l'échelle pose problème. Cette absence d'effet du choix des mappings sur les performances est confirmée par les commentaires des sujets. Ceux-ci déclarent utiliser principalement les variations, surtout rapides, des paramètres afin d'identifier les objets. Ils ne ressentent pas de différence d'efficacité entre les mappings.

La troisième expérimentation donne des temps de réponse plus longs surtout pour la deuxième partie, qui peuvent s'expliquer par l'utilisation de séquences sonores plus complexes. On observe dans la première partie, concernant des séquences sonores simples, une baisse significative des performances des combinaisons dès qu'un paramètre graphique perturbe un autre, comme la distance avec la taille et la transparence avec la luminance. Nous reviendrons sur cette nécessité d'indépendance dans

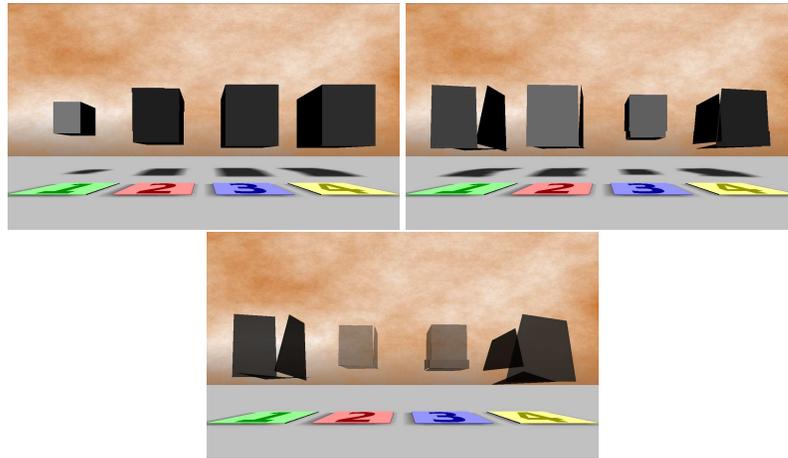


FIG. 3.17 – Capture d’écran de l’expérimentation sur les combinaisons avec deux, trois et quatre paramètres combinés.

nos conclusions sur les combinaisons de mappings. La deuxième partie ne montre pas de changements significatifs des performances d’identification, avec des séquences sonores complexes, lorsque l’on ajoute des mappings. Ceci ne permet pas de conclure à un impact ou une absence d’impact du nombre de mappings. Cependant, les problèmes de perturbation et d’échelles ayant plutôt tendance à réduire les performances, on peut imaginer que les performances s’amélioreraient ou du moins ne changeraient pas si ces problèmes étaient résolus. Ainsi, il serait possible de combiner plusieurs mappings pour des sources sonores, sans impacter les tâches d’identification. Lors de cette expérimentation, les sujets font remarquer les problèmes de perturbation, et ils expriment également des difficultés d’identification lorsque des variations de paramètres sonores ne sont pas représentées.

### Choix des mappings : Échelles et préférences

Quelles implications ont ces résultats et les travaux existants sur le choix des mappings ?

Tout d’abord le choix des mappings ne semble pas avoir d’impact sur les performances pour une tâche d’identification. Il serait évidemment important de les tester dans d’autres tâches (visualisation, manipulation). Les différences d’efficacité révélées par les résultats et les commentaires semblent plutôt provenir du choix des échelles. C’est donc principalement ce choix d’échelle qui s’avère important pour les mappings audiovisuels. Il devra évidemment s’appuyer sur les travaux effectués dans le domaine de la visualisation d’informations. Par exemple, des échelles de couleurs permettant une meilleure visualisation sont explorées par Zhang et al. [138].

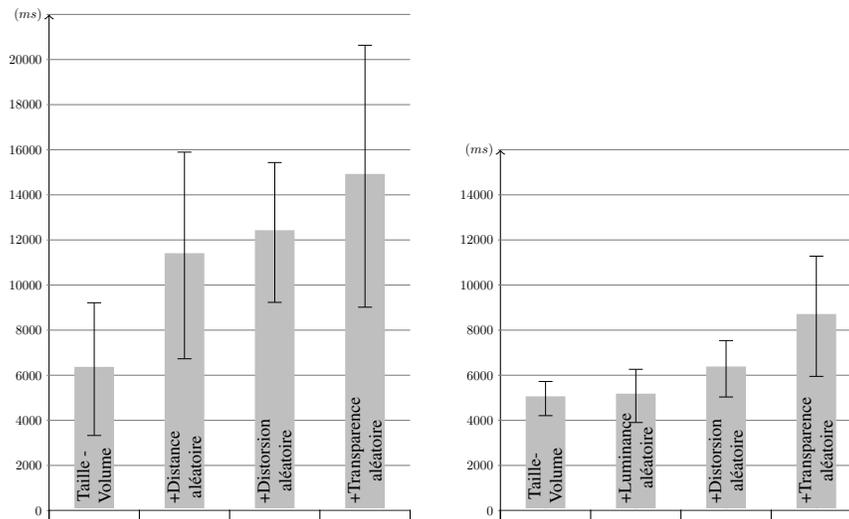


FIG. 3.18 – Troisième Expérimentation : Perturbations. Temps de réponse moyens et déviations moyennes avec la Distance (à gauche) puis la Luminance (à droite).

Si nous ne choisissons pas les mappings pour leurs performances, nous pouvons néanmoins nous appuyer sur les préférences des utilisateurs. Cela sera surtout intéressant si l'objectif est que les musiciens comprennent intuitivement les associations. Dans le cas où un instrument doit être appris, ces préférences sont au final moins importantes que le choix des échelles des paramètres. La première expérimentation conforte les résultats obtenus dans les travaux existants et confirme que les associations préférées sont celles liées à la pratique musicale (partitions, potentiomètres), à des analogies physiques (taille et distance/volume, taille/hauteur) ou à des principes logiques (transparence/volume, forme/timbre). Néanmoins, comme l'a montré Giannakis, les mappings perceptifs restent à favoriser car ils révèlent par exemple supérieurs à une représentation par sonogramme.

En plus du choix des échelles, trois pistes nous paraissent importantes à explorer :

- La notion de mappings relatifs et de mappings absolus :  
 Dans tous les tests effectués, la représentation par un paramètre graphique de la valeur d'un paramètre sonore est absolue. Ainsi pour deux objets audiovisuels 3D différents, une même couleur signifiera une même hauteur. Cependant, dans certains instruments, comme par exemple le *live-looping hiérarchique*, des effets s'appliquent sur des contenus audio complexes comportant déjà des variations. Ainsi un changement de hauteur appliqué sur une mélodie transpose toutes les notes. Dans ce cas, une même couleur pour deux objets 3D n'implique pas que leurs contenus ont la même hauteur. On peut alors parler de mappings relatifs. Il serait intéressant d'explorer les préférences et performances de ce

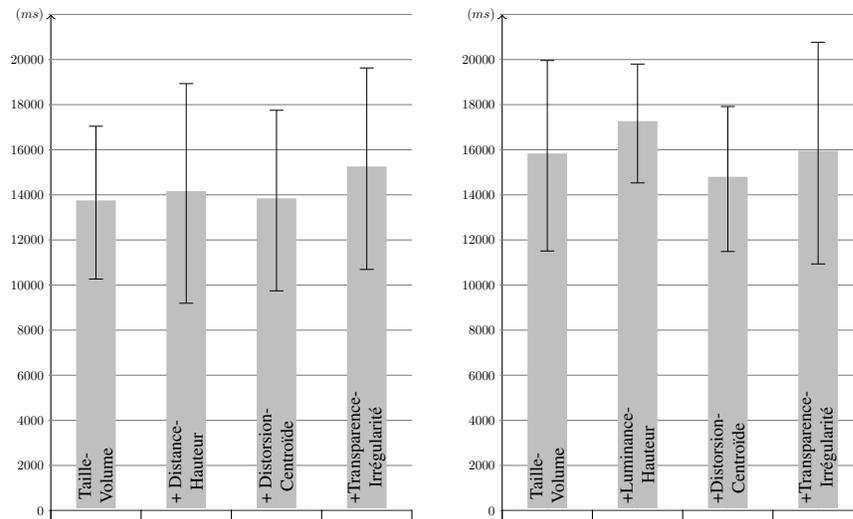


FIG. 3.19 – Troisième Expérimentation : Combinaisons. Temps de réponse moyens et déviations moyennes avec la distance (à gauche) puis la luminance (à droite) associés à la hauteur.

type de mappings, qui sont évidemment utilisés dès qu'un instrument opère sur des processus de haut niveau.

– La forme 3D :

Mis à part la distorsion, nos tests utilisateurs n'ont pas abordé l'utilisation de la forme 3D comme paramètre. Elle pourrait cependant servir à représenter des paramètres simples, par exemple en s'appuyant sur des formes 3D génératives [45]. Mais il est surtout intéressant de partir de formes 3D à plusieurs dimensions, ou de formes 3D composées afin de représenter des paramètres sonores comportant eux-mêmes plusieurs dimensions. Par exemple, comme nous le verrons dans le chapitre 5, nous pouvons utiliser un empilement d'anneaux pour représenter le spectre d'un processus sonore par la taille des anneaux, mais également d'autres paramètres par leur densité.

– Les mappings complexes : Si l'on rapporte les mappings utilisés aux types de mappings présentés dans la section 1.1, nous n'utilisons que des mappings de type one-to-one. Une piste à explorer serait donc la mise en œuvre de mappings one-to-many, voire many-to-many. On peut par exemple imaginer associer à la fois le volume et la hauteur à la taille de l'objet, avec des échelles différentes.

**Combinaison de mappings : Indépendance et Presets**

Bien que nos résultats ne montrent pas d'amélioration des performances par la combinaison de mappings, ils ne montrent pas non plus la chute de ces performances. Nous pouvons donc supposer que, comme c'est le cas pour la visualisation de données génériques, les mappings audiovisuels 3D peuvent être combinés pour la représentation de processus sonores, avec les avantages présentés pour les *widgets réactifs 3D*, moyennant les quelques contraintes suivantes.

Le premier point important pour les combinaisons de mappings, comme nous le montrent les résultats de la troisième expérimentation, est la nécessaire indépendance des paramètres perceptifs, évoquée notamment par Giannakis et Smith [55]. Ainsi, l'utilisation conjointe de variations de taille et de distance, ou de luminance et de transparence, brouille la perception des paramètres et donc perturbe les mappings. Les paramètres supplémentaires apportés par les environnements virtuels 3D permettent d'éviter beaucoup de conflits. Par exemple, la rotation sur l'axe Y peut être utilisée sans modifier la perception de la taille verticale d'un objet.

Un autre point à prendre en compte est la hiérarchie des paramètres visuels et sonores. Comme nous l'avons décrit plus haut, les paramètres perceptifs sont hiérarchisés au niveau du système visuel humain. Par exemple, des variations de luminance perturbent la perception de motifs de teintes, ce qui ne se produit pas dans l'autre sens. De la même façon, les variations de teinte altèrent la perception de la texture, tandis que les variations de texture n'affectent pas la perception de la teinte. Ainsi il est important de choisir les combinaisons de mappings en fonction de cette hiérarchie, notamment par ordre d'importance des paramètres sonores à visualiser.

Il serait tentant de laisser les utilisateurs modifier les mappings et leurs combinaisons. Cependant, étant donnée les contraintes que nous venons d'évoquer, il paraît plus pertinent de définir une seule ou un ensemble de configurations ou "presets" utilisables. Une interface telle que celle présentée sur la figure 3.20 peut alors être utilisée afin de parcourir et activer ces combinaisons de mappings.

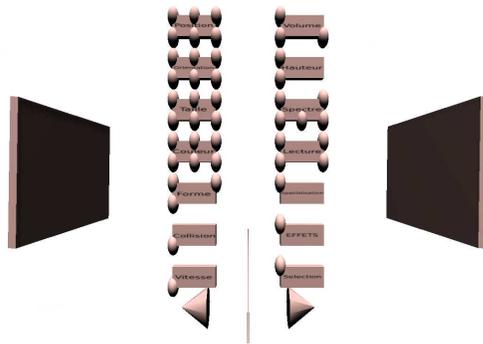


FIG. 3.20 – Interface de mapping : les connexions entre les paramètres graphiques (à gauche) et sonores (à droite) peuvent être enregistrées et rappelées sous forme de presets.

## Chapitre 4

# Interaction avec des environnements virtuels musicaux

*Percussion music is revolution. Sound and rhythm have too long been submissive to the restrictions of nineteenth century music. Today we are fighting for their emancipation. Tomorrow, with electronic music in our ears, we will hear freedom. At the present stage of revolution, a healthy lawlessness is warranted. Experiment must necessarily be carried on by hitting anything-tin pans, rice bowls, iron pipes-anything we can lay our hands on. Not only hitting, but rubbing, scraping, making sound in every possible way...What we can't do ourselves will be done by machines which we will invent. John Cage*

Dans ce chapitre, nous décrivons les techniques d'interaction musicale 3D que nous avons développées, notamment afin de manipuler les widgets réactifs 3D dans un environnement immersif. Dans la section 4.1 nous présentons *Piivert* [12], un nouveau dispositif pour l'interaction 3D immersive musicale. La section 4.2 définit les *tunnels* [9], éléments graphiques qui permettent de modifier les paramètres d'objets 3D.

### 4.1 Piivert : Percussion-based Interaction for Immersive Virtual Environments

#### 4.1.1 Motivation : Limitations des dispositifs d'interaction 3D

Comme nous l'avons vu dans le chapitre 3, les widgets réactifs 3D permettent de visualiser et de manipuler de manière cohérente plusieurs paramètres de plusieurs processus sonores. Les environnements immersifs combinant grand écran, stéréoscopie et suivi des mouvements de la tête, facilitent l'interaction avec ces widgets. En particulier,

des techniques telles que le rayon virtuel vont nous permettre de facilement accéder à n'importe quel widget dans l'environnement 3D. Le musicien va pouvoir pointer un widget réactif 3D, et ainsi manipuler le processus sonore associé. Cependant l'interaction 3D traditionnelle souffre de limitations en terme de précision temporelle et de retour haptique. Cette problématique ne se pose pas seulement dans le cas des widgets réactifs 3D, mais pour n'importe quel instrument immersif.

### **Limitations des Boutons/Joysticks**

Les dispositifs d'interaction portables utilisés dans ces environnements immersifs, et décrits dans le chapitre 1, reposent sur du suivi de position et d'orientation ainsi que sur des boutons ou joysticks. Ces caractéristiques se retrouvent dans le cas des nouveaux dispositifs de jeu tels que la Nintendo Wii-Remote. Les boutons et les joysticks ont une quantification très faible (1/8 bits) et le nombre de gestes possible est très restreint. Ces composants ne sont donc pas adaptés au contrôle musical qui nécessite une grande expressivité et donc une quantification suffisante. Par exemple, un bouton ne renvoyant qu'une valeur binaire ne peut pas reproduire la dynamique d'un geste de percussion.

### **Limitations du Suivi de Position / Interaction graphique**

D'un autre côté, le suivi de position 6 degrés de liberté (DDL), quel que soit le système utilisé (optique, électromagnétique, accéléromètres, ...), assure une plus grande quantification et donc une plus grande expressivité. Cependant, le contrôle graphique qui repose sur ce suivi souffre de certaines limitations pour une utilisation musicale.

#### **– Problèmes de résolution temporelle**

La première limitation vient de la latence inhérente aux systèmes de suivi et à l'interaction graphique 3D. En effet, les algorithmes nécessaires au calcul de la position et de l'orientation des dispositifs entraînent une latence due à leurs complexités et à leurs fréquences d'exécution. Ainsi, si un système de suivi optique envoie les informations à 60Hz et que le moteur physique est cadencé à 100Hz, on arrive très vite à une latence trop importante (surtout si l'on ajoute la latence du moteur audio) pour certains gestes musicaux tels que les gestes d'excitation instantanés. Ceci sera particulièrement sensible à des tempos rapides, les gestes finissant par se mélanger. A cette latence du système de suivi s'ajoute la latence due aux algorithmes d'interaction graphiques, par exemple pour la détection de collision entre par exemple le rayon virtuel et des objets 3D. Des techniques comme le déclenchement de sons à l'aide de baguettes virtuelles ne permettent donc pas de jouer des rythmes correctement. L'amélioration de la latence des systèmes de suivi est évidemment une voie à suivre mais les solutions ne seront pas accessibles (dans le temps et en terme de coût) dans un futur proche, il est donc nécessaire de trouver une méthode alternative.

Cependant, ce manque de précision temporelle concerne surtout les gestes d'excitation et de modulation, les gestes de sélection s'effectuant à une vitesse moindre. En effet, les gestes d'excitation, qui "produisent" les sons ne doivent pas être décalés temporellement. La latence entre le geste et le son, si elle est supérieure à la limite généralement admise de 10ms et d'autant plus s'il y a un retour haptique, devient perceptible et va affecter le jeu musical, notamment à des tempos rapides.

– Absence de retour haptique

La seconde limitation vient surtout du contexte de l'interaction 3D immersive. Le retour haptique est une composante importante du contrôle musical [99]. Par exemple, les instrumentistes apprennent généralement à jouer de leur instrument en s'appuyant sur le retour visuel, mais ils gagnent en expertise, et éventuellement deviennent virtuoses, en se reposant sur le retour haptique. Il est donc primordial de fournir ce retour haptique pour l'interaction musicale immersive.

Ce retour haptique pour l'interaction graphique peut s'appuyer soit sur un système de retours vibrotactiles soit sur un système de retour haptique actif. Cependant, pour profiter des possibilités d'immersion et de collaboration offertes par des systèmes s'appuyant sur de larges affichages stéréoscopiques, il est essentiel d'utiliser des dispositifs d'interaction portables, c'est à dire permettant au musicien de se déplacer librement. Les systèmes de retour haptiques actifs existant (bras mécaniques, câbles, ...) restreignent les déplacements des utilisateurs. Les retours vibrotactiles ne permettent pas, contrairement à un contact réel avec une surface, d'effectuer certains gestes musicaux nécessitant par exemple un rebond.

Encore une fois, ce problème rend l'utilisation de l'interaction graphique s'appuyant sur le suivi de mouvements 6DDL peu souhaitable.

## 4.1.2 Approche

### Séparation des gestes instrumentaux

Nous proposons de séparer l'interaction 3D avec les processus sonores en suivant les catégories de gestes instrumentaux définies par Cadoz [28] et que nous avons rappelés dans le chapitre 1. Nous pensons que cette séparation permet d'améliorer l'interaction avec des instruments immersifs en contournant les limitations de l'interaction graphique 3D pour certains gestes instrumentaux. Nous avons donc mis au point un dispositif d'interaction appelé *PIIVERT*, pour *Percussion-based Interaction for Immersive Virtual Environments*, auquel sont associées un certain nombre de techniques d'interaction. Ce dispositif est composé de marqueurs pour le suivi de la position et de l'orientation 3D ainsi que de capteurs de pression. Nous voulons ainsi séparer l'interaction graphique, s'appuyant sur le suivi de mouvements, de l'interaction musicale utilisant les capteurs de pression. Ainsi le traitement des gestes instrumentaux nécessitant une grande précision temporelle peut être complètement dissocié du traitement de l'in-

teraction graphique.

### **Gestes de sélection**

Les gestes de sélection sont effectués graphiquement en utilisant la technique du rayon virtuel, qui permet à l'utilisateur de pointer des objets dans l'environnement 3D. Le rayon est "attaché" au dispositif grâce au suivi de position, comme nous l'avons expliqué dans la section 1.3.4. Associée à l'affichage stéréoscopique et au suivi de la tête de l'utilisateur, cette technique se révèle efficace pour la sélection d'objets 3D peu éloignés, comme démontré par [100]. Les interfaces graphiques permettent d'accéder à un nombre potentiellement infini d'éléments graphiques. Ces éléments peuvent donner des informations visuelles riches sur les processus sonores qui leurs sont associés, ainsi que sur chacun des paramètres de ces processus. Contrairement aux gestes d'excitation qui ne peuvent pas être altérés ou retardés car cela entraînerait une modification du résultat musical, ces gestes de sélection peuvent de plus être "aidés", facilités, par les techniques d'interaction graphique, comme nous le précisons dans la section 4.1.4.

### **Gestes de modulation**

Les gestes de modulation peuvent aussi être effectués graphiquement. Ceci permet de bénéficier d'un retour visuel sur la modification des paramètres sonores en les associant, comme cela est fait pour les widgets réactifs 3D, avec des paramètres graphiques tels que la forme, la position dans l'espace, les matériaux. De plus les échelles de ces paramètres graphiques peuvent facilement être adaptées ce qui permet d'effectuer aussi bien des gestes de modulation discrets que des gestes continus. En acquérant de l'expertise, ces gestes de modulation peuvent également être effectués en utilisant des capteurs avec retour haptique.

### **Gestes d'excitation**

Pour tous les gestes instrumentaux, et notamment les gestes d'excitation, nécessitant un retour haptique et une grande précision temporelle, les capteurs utilisés dans les contrôleurs matériels traditionnels, tels que les potentiomètres, les capteurs de pression ou de vibration, sont beaucoup plus adaptés que l'interaction graphique. Ils apportent en effet une grande fréquence d'échantillonnage et une quantification plus élevée, combinées à un retour haptique passif. En utilisant ces capteurs, les musiciens peuvent effectuer des gestes experts : expressifs et précis.

### **Gestes de contrôle**

En plus des gestes d'excitation, les capteurs peuvent permettre de déclencher des actions qui nécessitent une intégration précise avec des événements musicaux, comme par exemple l'enregistrement de boucles musicales, le déclenchement d'animations et ainsi de suite.

## Percussions

Parmi tous les types de gestes musicaux qui peuvent être captés, les gestes de percussion sont intéressants pour plusieurs raisons. Ils sont tout d'abord faciles à analyser à l'aide de capteurs de pression, avec une grande précision et une quantification élevée. De plus, un grand nombre de techniques de percussion existent : frappes (main, doigts), frottements, pressions pour moduler le son ... Finalement, ces gestes possèdent un grand nombre de paramètres permettant de donner une sémantique aux gestes de contrôle. Ce sont donc ces gestes musicaux que nous allons extraire des capteurs, afin d'effectuer les actions nécessitant une grande précision et expressivité.

### 4.1.3 Dispositif

Le dispositif d'interaction Piivert est bâti sur cette répartition des gestes musicaux. Il associe donc des marqueurs de suivi 6DDL pour l'interaction graphique et des capteurs de pression pour les gestes de percussion.

## Ergonomie

Les deux premiers prototypes de Piivert que nous avons développés étaient respectivement tenu dans la main et attaché à l'avant-bras, comme le montre la figure 4.1. L'inconvénient du premier était qu'il restreignait le nombre de doigts utilisables, puisqu'au moins deux servaient à tenir le dispositif. Le deuxième en revanche laissait tous les doigts libres mais, le dispositif étant attaché à l'avant-bras, les mouvements du poignet n'étaient pas utilisables pour diriger le rayon virtuel, ce qui réduisait fortement la précision et la vitesse de l'interaction et augmentait la fatigue à l'utilisation.

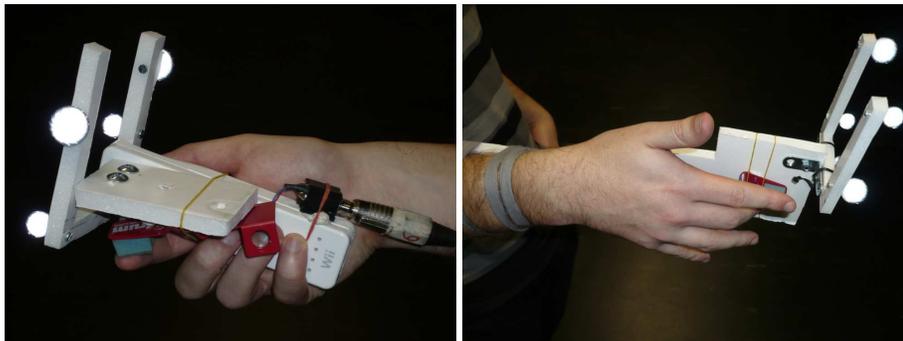


FIG. 4.1 – Premiers prototypes

Le prototype actuel est attaché à la main par une sangle semi-rigide. Il permet ainsi d'utiliser tous les doigts sans empêcher de contrôler les mouvements du rayon à l'aide du poignet. Comme visible sur la figure 4.2, le musicien est équipé d'un dispositif dans chaque main, ainsi que d'une boîte contenant la carte d'acquisition des capteurs fixée

à sa ceinture. Cette ergonomie est fortement inspirée du dispositif "The Hands" de Michel Waisvisz [125].

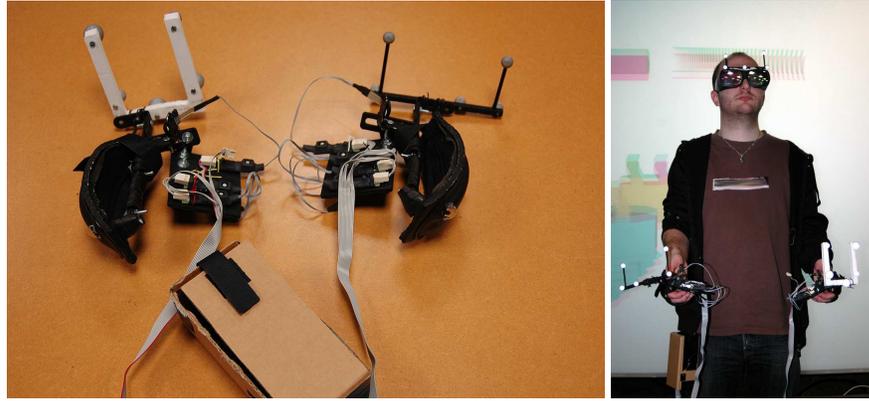


FIG. 4.2 – Prototype actuel de Piivert

### Suivi de mouvements

Le suivi de mouvements à 6 DDL de Piivert, qui va être utilisé pour les gestes de sélection et de modulation, est réalisé grâce à des marqueurs et aux caméras infrarouges du système A.R Tracking DTrack<sup>1</sup>. Les positions et orientations sont calculées par l'application DTrack sur ordinateur dédié à partir des images transmises. Ces données sont ensuite envoyées à l'application 3D qui gère l'environnement et les rayons virtuels en utilisant le protocole de communication VRPN [114]. La fréquence d'échantillonnage maximum de ce système de suivi de mouvements est de 60fps.

### Capteurs de pression

Des capteurs FSR (Force Sensitive Resistors) sont utilisés pour acquérir les gestes de percussion, afin de pouvoir détecter à la fois des pressions, et des frappes [115]. Comme on peut le voir sur la figure 4.3, ces capteurs sont placés sous les index, majeur et annulaires de chaque main. Les capteurs pour les pouces sont situés de manière à ce qu'ils puissent être frappés avec la tranche du pouce. Le petit doigt n'est pas pour l'instant utilisé, par contrainte technique. La capture est effectuée à l'aide d'une carte Interface-Z<sup>2</sup> qui transmet les données à une carte son en utilisant des messages et un câble Musical Instrument Digital Interface (MIDI)<sup>3</sup>. La carte envoie un maximum de 1500 données par seconde. Si les huit capteurs sont utilisés simultanément, cela donne approximativement 187Hz donc une latence de 5ms. Chaque donnée est encodée sur 12 bits, i.e 4096 valeurs, et est obtenue en mixant deux message MIDI Control Change de 7 bits chacun. La carte est connectée à une carte usb externe et les messages MIDI sont transmis à l'application audio en utilisant le driver et le séquenceur ALSA<sup>4</sup>.



FIG. 4.3 – Capteurs FSR sous l'index, le majeur et l'annulaire (gauche). Capteur FSR sur le côté du pouce (droite)

---

<sup>1</sup><http://www.ar-tracking.de>

<sup>2</sup><http://www.interface-z.com/>

<sup>3</sup><http://www.midi.org/>

<sup>4</sup><http://www.alsa-project.org>

### Implémentation

Une implémentation complète d'un instrument immersif utilisant Piivert est représentée sur la figure 4.4. Les deux flux de données (suivi de mouvements et capteurs) sont complètement séparés. Lorsque qu'un geste de sélection ou de modulation est effectué, l'application graphique applique les changements à l'environnement 3D et en informe l'application audio. Lors du geste d'excitation suivant, l'application audio peut donc directement déclencher le processus sonore, sans "attendre" l'application graphique. Ainsi les imprécisions du suivi de mouvements ne perturbent pas les contrôles musicaux.

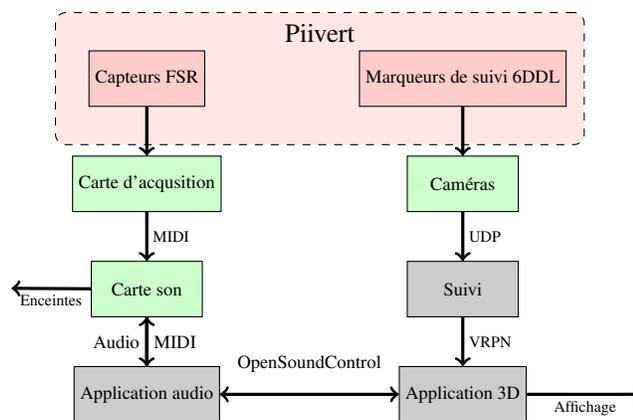


FIG. 4.4 – Implémentation d'un instrument immersif utilisant Piivert

Le flux audio de sortie est transmis par le serveur de sons JACK<sup>5</sup>, ajoutant seulement une latence de 5ms à la latence des capteurs. Les gestes d'excitation peuvent donc être effectués sans latence perceptible, ce qui ne serait pas le cas s'ils s'appuyaient sur le suivi de mouvements, ainsi que le montre la figure 4.5.

#### 4.1.4 Techniques d'interaction

Après avoir détaillé le dispositif matériel et son implémentation, nous allons maintenant décrire les gestes de percussion extraits des capteurs, qui sont utilisés pour les gestes requérant une grande précision temporelle et une grande expressivité. Ils pourront être employés comme gestes d'excitation mais également comme gestes de contrôles pour des actions nécessitant une intégration précise dans des événements musicaux. Nous décrirons ensuite les techniques d'interaction graphique utilisées pour les gestes de sélection et de modulation.

<sup>5</sup><http://jackaudio.org/>

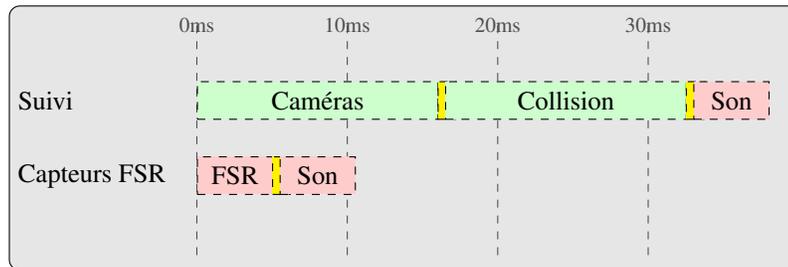


FIG. 4.5 – Latences (approximatives) de traitement des données des capteurs FSR et du suivi de mouvements (en jaune les transmissions réseaux et MIDI). La latence inhérente au suivi de mouvements le rend inutilisable pour des gestes musicaux tels que des excitations instantanées.

### Gestes de percussion

Les gestes de percussion sont extraits des données des capteurs FSR. Nous commençons par identifier des gestes de bas niveau puis leurs combinaisons en gestes de haut niveau. Dans l'implémentation actuelle, chaque geste de percussion déclenche une fonction de rappel prenant en argument un évènement *piivert\_event\_\**, qui contient le type de geste, le ou les doigts utilisés et la vélocité du geste.

#### Gestes de bas niveau :

A partir des données extraites pour chaque capteur, c'est à dire chaque doigt, nous pouvons extraire 3 gestes de percussion de bas niveau.

- Le premier geste est le "Hit", c'est à dire un coup simple, défini comme un *piivert\_event\_hit*. Il est détecté comme un saut dans les données du capteur de 0 à une valeur située au-dessus d'un seuil que nous avons fixé à 100, sur une durée de moins de 50ms. Ces deux seuils ont été définis de manière empirique suite à plusieurs expérimentations. La valeur du "Hit", c'est à dire la vélocité, est le premier maximum local qui suit le saut. Cette valeur, comme celle des autres gestes de bas niveau, est ensuite normalisée. Un exemple de données correspondant à un geste "Hit" est donné sur la figure 4.6.
- Le deuxième geste est le "Pressure", c'est à dire une pression lente. Ce geste est détecté à chaque variation lente des données du capteur, par exemple si le geste n'atteint pas suffisamment rapidement le seuil de déclenchement du "Hit" ou encore pendant les variations de pression suivant un "Hit". Durant ce geste, chaque valeur du capteur est envoyée comme valeur d'un évènement *piivert\_event\_pressure*.
- Finalement, un évènement *piivert\_event\_release* est détecté quand la valeur du capteur retourne à 0, que ce soit après un geste "Hit" ou "Pressure". La valeur de ce geste est la dernière valeur supérieure à 0 avant le saut à la valeur 0.

Nous utilisons une notation simple pour ces gestes, analogue à une partition ins-

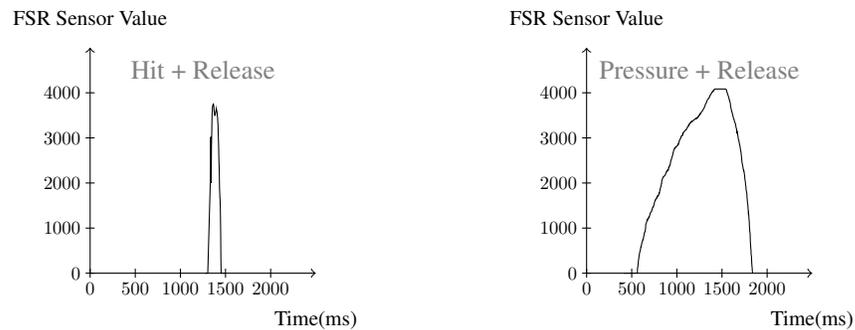


FIG. 4.6 – Exemple de données d'un capteur FSR résultant des gestes "Hit" et "Release" à gauche, et des gestes "Pressure" et "Release" à droite.

strumentale. Ceci nous permet de facilement décrire comment réaliser les différents contrôles définis dans un instrument utilisant Piivert. La notation des gestes de bas niveau est présentée sur la figure 4.7.

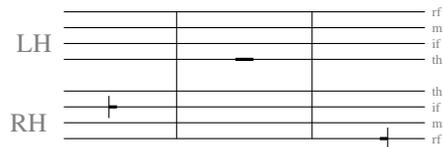


FIG. 4.7 – Notation des gestes de bas niveau ( RH : main droite, LH : main gauche, th/if/mf/rf : doigts ). De gauche à droite : Hit, Pressure, Release

De par leur simplicité et instantanéité, les gestes de bas niveau sont plutôt destinés à être utilisés comme gestes d'excitation. Pour chaque doigt, ces gestes peuvent de plus être associés à des techniques de synthèse ou à des paramètres différents.

Cependant, on peut également utiliser ces gestes de bas niveau pour des actions non musicales. Par exemple, les gestes "Pressure" du pouce peuvent servir à saisir les objets 3D.

Gestes de haut niveau :

Les gestes de haut niveau sont détectés comme des combinaisons de gestes de bas niveau séparés de moins de 50ms. Dans notre implémentation actuelle, 3 types de gestes sont détectés.

- L'évènement *piivert\_event\_flam* décrit le geste "Flam" qui est une succession de deux "Hit".
- Une séquence de trois "Hit" est un "Three strike roll" et sera déclaré comme un *piivert\_event\_roll3*.

- Finalement, une séquence de quatre "Hit", aussi appelé "Four strike roll", correspond à un évènement *piivert\_event\_roll4*

En plus de ces trois gestes composés de successions de "Hit", des combinaisons de "Hit" et de "Pressure" peuvent être utilisées. Ainsi, Piivert permet de reproduire les gestes des percussionnistes qui appuient sur une peau tout en la frappant afin d'en modifier la hauteur. Tous ces gestes peuvent être effectués avec les doigts d'une seule main mais également avec des doigts de chaque main. Les notations pour ces gestes sont présentées dans la figure 4.8.

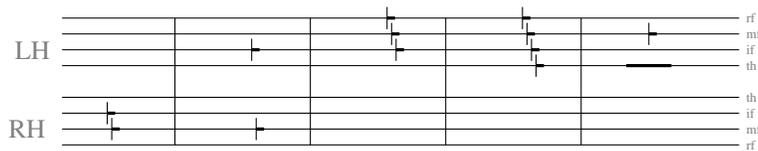


FIG. 4.8 – Notation des gestes de haut niveau. De gauche à droite : Flam à une main, Flam à deux mains, Three strikes Roll, Four strikes Roll, Combinaison de "Hit" et de "Pressure"

Ces gestes de haut niveau s'intègrent naturellement aux gestes de bas niveau en constituant des rythmes. Ceci permet de les utiliser pour des actions musicales nécessitant une grande précision temporelle. De plus, à l'inverse des boutons généralement utilisés sur les dispositifs d'interaction 3D, les gestes de haut niveau possèdent plusieurs paramètres. Tout d'abord, les doigts utilisés sont identifiés comme paramètre de l'évènement. Ceci permet à la fois de prendre en compte quels doigts appartiennent au geste, mais également dans quel ordre ils apparaissent. Ce paramètre de direction peut donner un sens à certaines actions. Par exemple, un sens pour déclencher une animation, et l'autre pour l'arrêter. La bimanualité peut également être utilisée comme un paramètre, la même action pouvant par exemple se répercuter sur les deux rayons (un par main). Ces gestes de haut niveau ont également un paramètre de vélocité (toujours sur 12bits), défini comme la valeur du dernier geste de bas niveau, ainsi qu'un paramètre de durée (10-50ms pour un flam, 20-100ms pour un three strike roll et 20-150ms pour un four strikes roll).

#### Conflits :

Mélanger des gestes de haut niveau avec des séquences de gestes de bas niveau pose cependant certains problèmes. Des conflits peuvent en effet apparaître lorsque des "Hit" sont utilisés pour déclencher des sons. Il est parfois préférable d'empêcher ces "Hit" de produire du son lorsqu'ils sont effectués dans le cadre de gestes de haut niveau, dont les résultats peuvent être non-musicaux. Nous proposons différentes stratégies afin d'éviter ces conflits. La première est de définir un doigt de départ pour tous les gestes de haut niveau, afin de court-circuiter le geste effectué avec ce doigt ainsi que tous les gestes de bas niveau qui vont suivre et qui composeront le geste final. Une autre stratégie est de laisser passer le premier geste de bas-niveau mais d'empêcher les sui-

vants s'ils constituent un geste de haut-niveau. Dans ce cas, si le geste est correctement inséré dans un rythme, le geste de haut niveau ne perturbera pas la séquence musicale en cours.

Le système de fonction de rappel que nous proposons pour les gestes de percussion pourra permettre aux développeurs d'instruments de définir leur propres stratégies.

### **Interaction graphique pour les gestes de sélection et de modulation**

#### Rayon Virtuel :

Les gestes de sélection sont effectués graphiquement. Notre implémentation actuelle utilise la technique du rayon virtuel. Un rayon virtuel (pour chaque main), rigide, est dirigé par le Piivert, comme on peut le voir sur la figure 4.9. La base du rayon est positionnée à l'extrémité du dispositif, afin de donner l'impression qu'il sort de ce dernier. Ceci permet par exemple au musicien de pointer directement des composants graphiques 3D correspondant à des processus sonores. Cette technique de sélection est très souvent utilisée pour l'interaction 3D, car elle est efficace pour la sélection rapide d'objets proches dans un environnement virtuel, ce qui a été démontré par Poupyrev et al. [100]. Pour des objets plus éloignés, la précision diminue, car l'écart de position d'un point du rayon lorsque l'orientation du dispositif est modifiée dépend logiquement de la distance de ce point par rapport à la base du rayon. Différentes techniques ont été développées pour remédier à ce problème, comme par exemple l'utilisation d'un cône de sélection. Pour plus de simplicité, nous nous limitons dans un premier temps à la sélection et manipulation d'objets audiovisuels 3D peu distants de l'utilisateur.

Cependant, l'enchaînement de notes à un tempo élevé peut nécessiter de sélectionner rapidement de petits objets 3D éloignés les uns des autres, ce qui va poser problème même lorsque ces objets sont proches. Afin de préserver la précision temporelle et spatiale du rayon virtuel, nous proposons plusieurs techniques qui facilitent ces gestes de sélection graphiques.

#### Boîtes englobantes :

Nous définissons tout d'abord des boîtes englobantes étendues par rapport aux objets 3D, représentées sur la figure 4.9. La détection de collision n'est plus réalisée entre le rayon et les objets mais entre le rayon et ces boîtes englobantes. L'objet est donc activé dès que le rayon s'en approche suffisamment. Ceci permet de minimiser les mouvements nécessaires pour passer d'un objet 3D à un autre en préservant la taille de ces objets. Les musiciens n'ont alors plus besoin de pointer exactement sur les objets afin de déclencher les processus sonores associés par des gestes d'excitation. Le passage d'un objet audiovisuel à un autre peut donc être beaucoup plus rapide. De même, la sélection d'objets 3D plus éloignés est facilitée.

Une autre stratégie possible serait de faire sauter automatiquement le rayon virtuel à l'objet le plus proche comme proposé par [41].

Quelle que soit la méthode choisie, on remarque que la séparation, entre les gestes d'excitation réalisés avec les capteurs et les gestes de sélection réalisés graphiquement,

permet de faciliter ces derniers sans perturber les premiers.

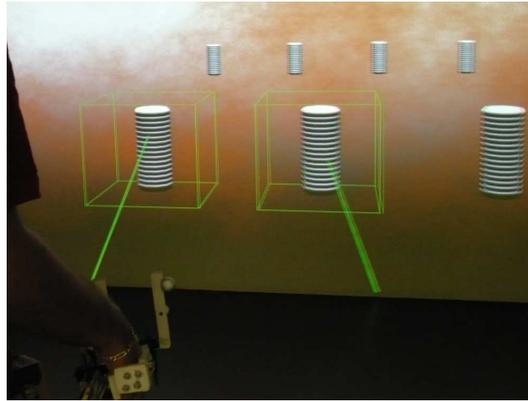


FIG. 4.9 – Sélection avec les rayons virtuels et boîtes englobantes adaptées.

#### Rayon Multiple :

Bien que la manipulation du rayon virtuel soit aisée, son utilisation pour jouer des séquences musicales rapides nécessite tout de même un certain apprentissage. Pour faciliter le jeu de mélodies ou rythmes complexes, nous pouvons profiter de la présence sur Piivert d'un capteur FSR pour chaque doigt. Nous proposons de séparer le rayon virtuel en deux ou trois rayons, et d'affecter un rayon à chaque doigt. Un mode sélection permet tout d'abord de choisir de un à trois objets 3D, puis le rayon est séparé en fonction du nombre d'objets sélectionnés. Chaque objet 3D est alors attaché au rayon correspondant. Par exemple le rayon peut être séparé en trois, comme sur la figure 4.10, afin d'avoir sous l'index un objet associé à un son de caisse claire, sous le majeur un son de grosse caisse et sous l'annulaire un son de cymbale. Des gestes "Hit" permettent alors de jouer un rythme de batterie avec ces trois doigts, comme ils serait possible de le faire sur un clavier. Cependant, contrairement à un contrôleur matériel, les gestes de sélection graphique vont donner accès à un nombre potentiellement infini de processus sonores.

#### Rayon vibrant :

En plus de ces deux techniques d'aide aux gestes de sélection, nous utilisons une méthode pour adapter la technique du rayon virtuel à l'interaction musicale, le "Rayon vibrant", représenté en figure 4.11. Cette technique consiste à représenter les gestes d'excitation extraits par les capteurs comme des vibrations se propageant sur le rayon virtuel. Un geste "Hit" déclenche par exemple une impulsion unique qui se propage jusqu'à l'objet, tandis qu'un geste "Pressure" forme une sinusoïde dont l'amplitude est définie par la valeur du geste.

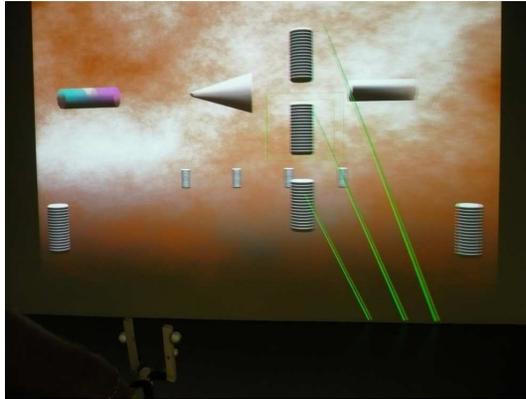


FIG. 4.10 – Rayon multiple. Ici le rayon est séparé en trois et permet donc de saisir et de manipuler 3 objets simultanément.

Dans le cas du "Rayon multiple", chaque capteur envoie les vibrations uniquement au rayon correspondant.

Ce retour visuel nous permet d'améliorer l'intégration des gestes musicaux dans l'environnement virtuel et ainsi de favoriser l'immersion du musicien.

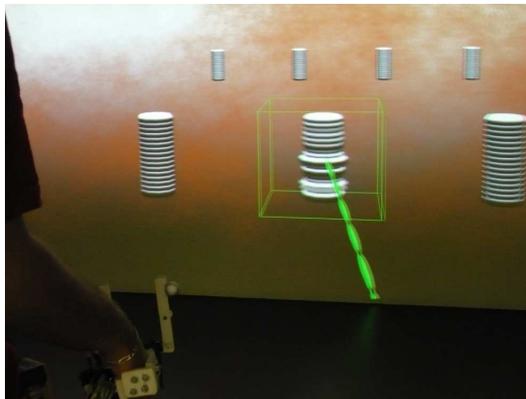


FIG. 4.11 – Le "Rayon vibrant" permet de visualiser les gestes d'excitation, ici un geste "Pressure"

#### Modulation :

Nous définissons les gestes de modulation comme les modifications des paramètres graphiques des objets 3D. Ces paramètres graphiques sont en effet associés aux paramètres sonores, comme nous l'avons défini pour les *widgets réactifs 3D* dans le

chapitre 3. Plusieurs possibilités existent alors pour modifier ces paramètres, qu'ils correspondent à la position, à l'orientation, à la forme, ou bien au matériau des objets 3D. Nous ne rentrerons pas ici dans les détails, une technique particulière pour cette modulation graphique étant présentée dans la section 4.2.

Avec Piivert, nous pouvons exciter les objets audiovisuels avec les capteurs tout en les modulant graphiquement. En combinant ceci avec la technique du "Rayon multiple", nous pouvons exciter et moduler en parallèle jusqu'à 3 objets avec chaque main, comme présenté sur la figure 4.12. Cette combinaison de gestes instrumentaux serait beaucoup plus compliquée à réaliser avec des contrôleurs matériels, notamment à cause des limitations en termes de sélection et de manipulations. En effet, il est par exemple impossible d'exciter et de moduler un processus sonore avec un seul doigt sans la séparation de ces gestes et les retours visuels suffisants.

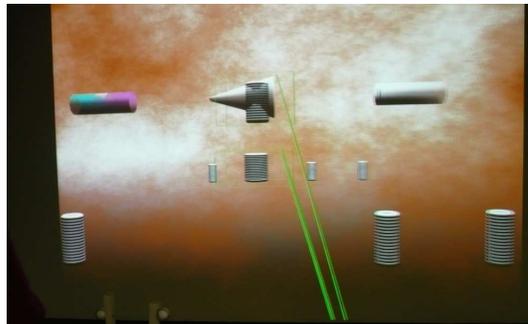


FIG. 4.12 – Geste de modulation couplé à la technique du "Rayon multiple"

### 4.1.5 Évaluation de Piivert

#### Hypothèse

Afin d'évaluer la pertinence de notre approche, c'est à dire la séparation de l'interaction 3D musicale en suivant les gestes instrumentaux, nous comparons Piivert avec la technique des "Baguettes virtuelles" dans le cas d'une tâche de déclenchement de son simples. La technique des "Baguettes virtuelles" est similaire aux contrôles gestuels couramment utilisés par exemple avec la Nintendo Wii Remote [18].

Notre hypothèse est que la séparation des gestes de sélection et d'excitation, les uns étant effectués graphiquement et les autres grâce aux capteurs de pression, améliore la précision temporelle et réduit le taux d'erreurs pour cette tâche particulière.

#### Conditions expérimentales.

L'expérimentation a été mise en place en utilisant le logiciel Pure-Data <sup>6</sup> pour les parties de contrôle et audio, et un logiciel s'appuyant sur la bibliothèque OpenGL <sup>7</sup>

<sup>6</sup><http://puredata.info>

<sup>7</sup><http://www.opengl.org>

pour le rendu de l'environnement 3D. Le suivi de mouvements 6DDL de Piivert est effectué à l'aide de caméras infrarouge A.R Tracking dont les données sont analysées par un ordinateur dédié qui envoie ensuite la position et l'orientation du dispositif à l'ordinateur de rendu par le protocole VRPN. Les collisions entre le rayon virtuel et les objets 3D sont envoyées par l'ordinateur de rendu à l'ordinateur "audio" en utilisant le protocole OpenSoundControl. Cet ordinateur "audio" est utilisé pour le rendu sonore et la détection des gestes dans les données envoyées par les capteurs. Le rendu sonore est effectué grâce à Pure-Data. La diffusion est réalisée par une carte son externe M-Audio FastTrack Pro, et deux enceintes actives M-Audio Studiophile BX5 placées à approximativement 4m des sujets.

Les sujets de cette expérience sont 16 musiciens d'un niveau instrumental allant de moyen à très bon, tous droitiers, 15 hommes et une femme, dont l'âge varie entre 23 et 41 ans. Seul l'un d'entre eux a une expérience de l'interaction 3D et de la réalité virtuelle, mais elle reste faible. Ils sont équipés de lunettes stéréoscopiques passives et leur tête est suivie (toujours grâce au système A.R. Tracking) afin d'adapter l'affichage de l'environnement 3D à leur position et ainsi les immerger correctement. L'affichage est réalisé à l'aide de deux vidéo-projecteurs (un pour chaque œil) utilisant la technologie Infitec<sup>8</sup>. Les sujets se tiennent à approximativement 4 mètres de l'écran et tiennent un dispositif Piivert dans leur main droite. Ces conditions sont représentées sur la figure 4.13.

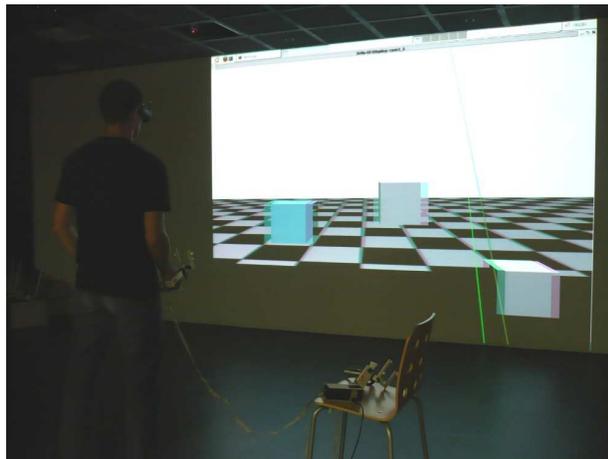


FIG. 4.13 – Conditions expérimentales.

### Procédure

La tâche consiste à rejouer une séquence musicale utilisant trois sons simples en sélectionnant et excitant trois objets 3D auxquels sont associés les sons. Pour la tech-

<sup>8</sup><http://www.infitec.net>

nique "Baguettes virtuelles", les sujets déclenchent les sons en tapant sur les objets 3D avec le rayon virtuel, comme ils le feraient avec une baguette de batterie et un fût. Pour la technique Piivert, les sujets doivent sélectionner l'objet 3D en positionnant le rayon dans la boîte englobante puis effectuer un geste "Hit" avec un doigt pour déclencher le son. Pour cette technique, les boîtes englobantes adaptées, présentées dans la section 4.1.4, sont utilisées. Les sujets peuvent donc sélectionner l'objet sans être exactement dessus, et ainsi minimiser leurs mouvements. Les trois objets 3D sont situés sur le même plan, c'est à dire à la même distance de l'utilisateur sur l'axe Z.

Les deux techniques sont testées dans un ordre aléatoire pour chaque sujet, et à deux tempos différents, 120 BPM (beats per minute) et 170 BPM. Les tempos ont été choisis afin de ne pas être trop lents ni trop rapides. Lors des tests, un son additionnel est joué sur tous les temps, comme un métronome. Ceci permet d'aider les sujets à se synchroniser avec le tempo lent. Chaque tempo comporte 4 essais. Chaque essai est composé de 4 séquences, jouées dans un ordre aléatoire, d'une à quatre notes disposées sur 4 temps. Une séquence d'entraînement est ajoutée lors d'un changement de technique, afin de réduire l'effet d'apprentissage. Les séquences sont les mêmes pour chaque sujet et ont été générées aléatoirement comme combinaisons des 3 sons/notes.

Durant une séquence, la séquence de notes est jouée une première fois. Quand une note est déclenchée, l'objet qui lui est associé change de couleur simultanément. La deuxième fois que la séquence jouée, les sujets doivent déclencher les objets en se synchronisant avec elle.

Nous enregistrons alors l'exactitude temporelle (en ms) des déclenchements, c'est à dire la moyenne des écarts entre les notes de la séquence et les déclenchements des sujets. Nous enregistrons également les déclenchements en trop ou manquants, qui constituent le taux d'erreurs.

## Résultats

Les figures 4.14 et 4.15 montrent les données obtenues à l'issue des expérimentations.

Les résultats d'une analyse de variance (ANOVA) à deux facteurs ne montrent aucun effet principal de la technique et du tempo sur la précision temporelle. Cependant, l'effet d'interaction étant significatif ( $F_{(1,15)} = 6.655, p < 0.05$ ), un test t apparié a été effectué sur les résultats pour chaque tempo. Les résultats de l'analyse pour le tempo lent ne montrent pas de différences significatives. Pour le tempo rapide, une différence significative apparaît entre les techniques ( $T_{15} = 3.737, p < 0.05$ ), Piivert étant plus précis que les "Baguettes Virtuelles".

Une différence significative ( $F_{(1,15)} = 7.484, p < 0.05$ ) entre les techniques est révélée par une ANOVA à deux facteurs sur le taux d'erreurs. Cependant la différence entre les tempos et l'effet d'interaction ne sont pas significatifs. Encore une fois, des test t de Student sont effectués pour les tempo lent et rapide. Les résultats montrent qu'il y a significativement moins d'erreurs pour Piivert au tempo lent ( $T_{15} = 2.35, p < 0.05$ ) et au tempo rapide ( $T_{15} = 2.239, p < 0.05$ ).

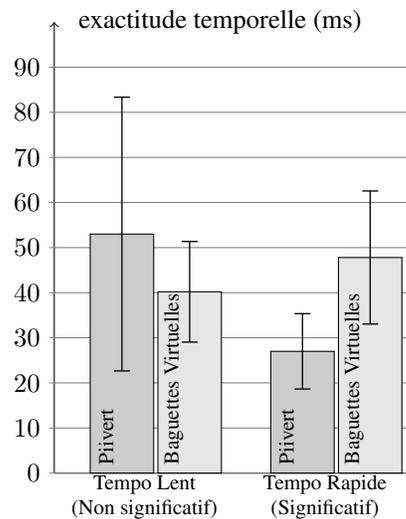


FIG. 4.14 – Précision temporelle de Piivert et des Baguettes Virtuelles à tempo lent puis rapide.

### Analyse et Discussion

Au tempo lent, *Piivert* donne une précision temporelle moins bonne que les "Baguettes Virtuelles", sans que la différence soit significative. Nous pensons que cela provient de la difficulté éprouvée par les sujets pour s'habituer à son utilisation, ce qui a été confirmé par les commentaires. En effet, le geste effectué pour les "Baguettes virtuelles" leur semblait plus naturel que celui de pointer avec le rayon puis de déclencher avec les capteurs. Un autre problème évoqué est l'utilisation même des capteurs, les sujets devant s'habituer à la course nécessaire pour les toucher avec leurs doigts. La lenteur du tempo peut également entraîner des hésitations au niveau des gestes d'excitation.

Au tempo rapide, la précision temporelle de *Piivert* devient significativement meilleure. Il semble que la précision des "Baguettes virtuelles" diminue à cause des mouvements plus larges et plus rapides à effectuer. Ici le suivi de mouvements et l'interaction graphique montrent leurs limites. Ce résultat est également à rapprocher de l'étude menée par Zhai et al. sur les groupes de muscles pour la manipulation 6DDL qui a démontré la précision des doigts par rapport aux poignets [137]. On peut supposer qu'à des tempos encore plus rapides la précision des "Baguettes Virtuelles" diminuerait à nouveau. Pour *Piivert*, ce tempo plus rapide semble réduire les hésitations car la continuité temporelle entre les déclenchements augmente. Il est légitime de penser que les sujets se sont de plus habitués à la technique. Plus globalement l'utilisation des boîtes englobantes adaptées et de *Piivert* permet aux sujets de réduire l'amplitude des gestes de sélection à effectuer tout en conservant la précision des gestes d'excitation.

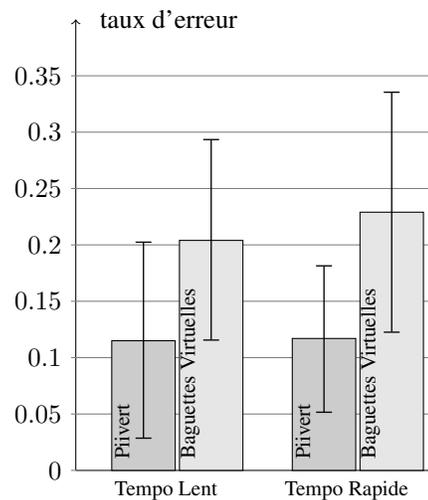


FIG. 4.15 – Taux d’erreur de Piivert et des Baguettes Virtuelles à tempo lent puis rapide.

Concernant les erreurs, Piivert devance les "Baguettes virtuelles" de manière significative pour les deux tempos. Avec cette dernière technique, les sujets ont en effet tendance à rater les objets ou à les taper plusieurs fois. Les commentaires révèlent que ce problème pourrait venir du manque de retour haptique lié à cette technique.

Dans cette expérimentation, les 3 objets 3D sont situés à la même profondeur. On peut supposer que la technique des "Baguettes virtuelles" perdrait en efficacité si ce n’était pas le cas. En effet, l’estimation du contact entre le rayon et les objets serait plus complexe à réaliser pour les sujets. Les performances de Piivert ne devrait cependant pas en être affectées, un pointage grossier étant suffisant pour les gestes de sélection.

Cette évaluation confirme en partie notre hypothèse selon laquelle la séparation des gestes de sélection (effectués graphiquement) et des gestes d’excitation (s’appuyant sur des capteurs de pression) améliore l’efficacité de l’interaction musicale avec des instruments 3D immersifs. Cependant, ces résultats ne sont valables que pour une tâche simple (déclenchement de sons associés à des objets 3D), et pour des utilisateurs novices. Il serait donc intéressant d’étendre cette expérimentation à d’autres tâches musicales, à d’autres techniques d’interaction, et d’étudier l’apprentissage de Piivert.

#### 4.1.6 Perspectives

L’objectif de Piivert est de permettre aux musiciens d’interagir de manière expressive et précise avec des instruments immersifs 3D, ce qui est en partie confirmé par l’évaluation que nous avons menée. Ils peuvent effectuer des gestes d’excitation et de

contrôle grâce aux capteurs de pression, et profiter de la richesse de l'interaction graphique 3D pour les gestes de sélection et de modulation.

Cependant, de nombreuses pistes restent à explorer.

– Ajout de capteurs :

Notamment, il pourrait être intéressant de rajouter des capteurs sur le dispositif, afin de permettre des gestes supplémentaires.

– Ajout de paramètres :

D'autres paramètres pourraient être pris en compte dans les gestes de haut niveau existants, par exemple la durée des gestes ou encore l'évolution des vitesses et les intervalles entre les gestes de bas niveau qui les composent.

– Interaction 3D :

Des techniques d'interaction graphique différentes du rayon virtuel pourraient être testées. Ce dernier souffre en effet de certaines limitations. La distance amplifiant les mouvements et donc les imprécisions du rayon, les objets éloignés sont plus difficiles à sélectionner et manipuler. L'occultation d'objets par d'autres est également un problème, seulement partiellement réglé par le suivi des mouvements de la tête. Finalement, des techniques telles que HOMER [22] permettent d'améliorer l'efficacité de la manipulation des objets 3D.

– Évaluation :

Une évaluation de Piivert par rapport à d'autres techniques serait intéressante après un certain temps d'apprentissage des sujets. De plus d'autres gestes, ou combinaisons de gestes, pourraient être évalués.

– Applications non musicales :

Les gestes de haut niveau de Piivert offrent à la fois une grande précision et de nombreux paramètres de contrôle. Il pourrait donc être utile dans des applications immersives non musicales dans lesquelles la temporalité est importante, par exemple le contrôle d'animations, la navigation ... Les gestes pourraient avantageusement remplacer les traditionnels boutons en apportant une sémantique aux actions. Au lieu d'utiliser plusieurs boutons, un utilisateur pourrait par exemple utiliser un "Flam" pour déclencher une animation, la vitesse du geste contrôlant la vitesse et la direction du geste contrôlant le sens de l'animation. De même, il serait intéressant d'intégrer les notations des gestes dans une interface, afin de facilement composer des gestes pour les actions des applications.

Finalement, il paraît important de distribuer la partie logicielle de Piivert sous la forme d'une bibliothèque afin de permettre son utilisation dans de nombreux instruments immersifs. Avec les instructions pour construire le dispositif et cette bibliothèque à priori simple à utiliser (une seule fonction de rappel à définir), de nombreux projets pourront profiter des possibilités offertes par Piivert.

## 4.2 Tunnels

### 4.2.1 Motivation

Nous avons vu dans la section précédente que les gestes de sélection et d'excitation pour les instruments 3D immersifs peuvent être réalisés de manière efficace avec *Piivert*. Les gestes de modulation, gestes qui permettent de modifier les paramètres d'un instrument, n'ont cependant pas été abordés. Comment manipuler les paramètres graphiques des *widgets réactifs 3D*, et donc les paramètres sonores qui leur sont associés ? Nous décrivons dans cette section la technique d'interaction graphique que nous proposons : les *tunnels*.

### 4.2.2 Principe

Les *tunnels* sont des composants graphiques 3D composés d'une centaine de cylindres creux horizontaux mis bout à bout, comme on peut le voir sur la figure 4.16. Ils sont associés à un ou plusieurs paramètres graphiques chacun ayant une ou plusieurs échelles. Les cylindres qui composent les *tunnels* prennent les valeurs des paramètres graphiques, suivant l'échelle utilisée. Ainsi un *tunnel* associé à la hauteur (taille sur l'axe Y) présente une variation de taille sur ses cylindres, et donc prendre la forme d'un cône.

Les *tunnels* modifient les paramètres graphiques des objets 3D qui les traversent, en leur affectant les valeurs correspondant aux échelles de modulation et à leur position (horizontale). Par exemple, un objet 3D passé à l'intérieur du *tunnel* précédent est rétréci lorsqu'il se trouve à l'extrémité gauche du *tunnel* et très étiré lorsqu'il est à l'extrémité droite.

Les paramètres graphiques des widgets réactifs 3D étant connectés aux paramètres sonores comme décrit dans le chapitre 3, les *tunnels* permettent de moduler les processus sonores associés.

Les tunnels sont définis par l'utilisateur dans des fichiers XML, comme présenté en annexe.

### Utilisation

Quand le musicien ne veut moduler qu'un seul objet audiovisuel 3D, il peut le saisir (par exemple en effectuant un geste "Pressure" avec *Piivert*) et le placer à l'intérieur d'un *tunnel*. L'activation de la modulation se fait donc verticalement si le musicien veut "sauter" à une valeur précise, ou horizontalement s'il souhaite commencer par une extrémité. La modification du paramètre s'effectue ensuite horizontalement. Si le musicien ressort l'objet du *tunnel* sans l'avoir relâché, le paramètre graphique reprend sa valeur initiale (celle qu'il avait avant d'entrer dans le *tunnel*). Si par contre il le relâche pendant qu'il est dans le *tunnel*, il garde la valeur correspondant à sa position une fois sorti. Ceci permet d'effectuer des modulations temporaires et de retrouver facilement la valeur initiale, ce qui peut être utile par exemple pour des variations de hauteur suivant une mélodie, ou pour revenir à une valeur n'étant pas présente dans l'échelle actuelle du paramètre. Cette modulation d'un

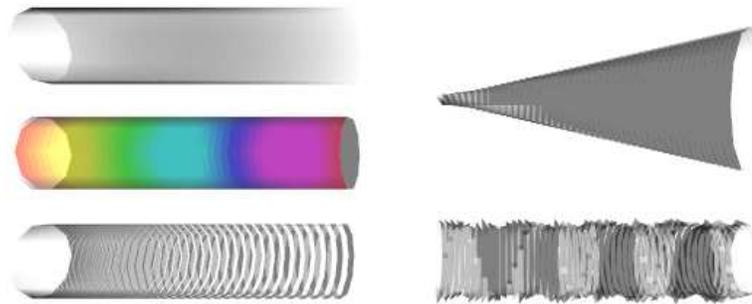


FIG. 4.16 – *Tunnels* à un seul paramètre (graphique/sonore) : (à gauche) Transparence/Caractère bruité, Teinte/Hauteur, Espace/Réverbération, (à droite) Taille/Volume, Vitesse de rotation/Tempo

seul objet peut être vue sur la figure 4.17 . Les *tunnels* utilisent le principe de l’affordance [57], leur forme suggère en effet leur utilisation. En l’occurrence, le fait qu’ils soient creux et allongés induit l’action de déplacer les objets à l’intérieur de ces tunnels.

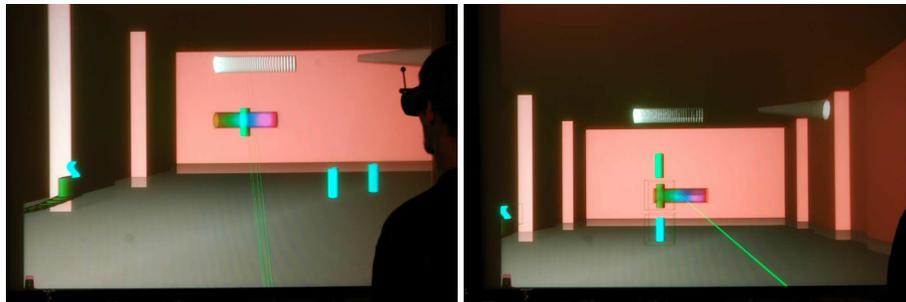


FIG. 4.17 – Modulation d’un seul objet en le passant à travers un *tunnel* (à gauche) et modulation quasi simultanée de plusieurs objets 3D en manipulant un seul *tunnel* (à droite)

Le musicien peut également vouloir moduler le même paramètre de plusieurs objets simultanément ou successivement. Il peut donc saisir un *tunnel* et le déplacer sur les objets 3D, comme présenté sur la figure 4.17 Si ces derniers sont alignés verticalement, cela permet d’affecter la même valeur à plusieurs objets 3D rapidement, par exemple pour transposer une série de processus sonores en un seul geste. S’ils sont alignés horizontalement, le musicien peut réaliser certains effets, tels que des “fade-out” (diminution du volume) sur plusieurs processus.

### Échelles

Les paramètres des *tunnels* peuvent comporter plusieurs échelles. Ceci permet d'avoir à la fois des modulations continues et des modulations discrètes. Les échelles continues sont définies par une valeur de début et une valeur de fin. Une interpolation linéaire est ensuite appliquée pour obtenir les valeurs du paramètre. Les échelles discrètes sont elles définies par un vecteur de valeurs du paramètre. Dans le cas d'un mapping avec la hauteur d'un processus sonore, des numéros de notes (0-48) peuvent également être utilisés pour définir l'échelle. Elles sont converties en valeurs brutes du paramètre.

La figure 4.18 montre plusieurs échelles possibles pour un même paramètre.

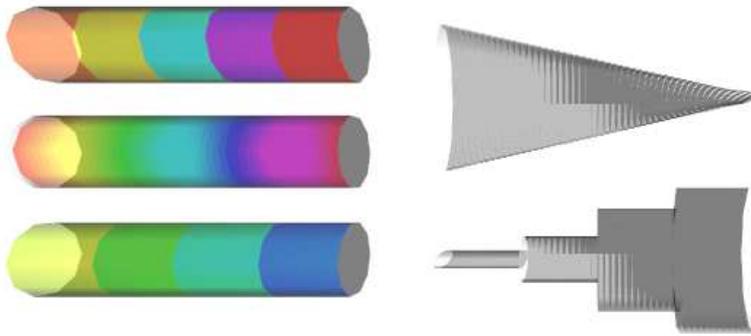


FIG. 4.18 – Plusieurs échelles pour la teinte (gauche) et la taille (droite).

Le passage d'une échelle à une autre peut se réaliser en sélectionnant le *tunnel* avec Piivert et en effectuant un geste "Hit".

### Résolution

Puisque la valeur de paramètre affectée à l'objet 3D dépend de sa position dans le *tunnel*, plus un *tunnel* est long, plus la résolution des paramètres est élevée. En effet, comme le montre le schéma 4.20, puisque la précision de positionnement des objets 3D est fixe (liée à la précision du suivi 6DDL), étirer un *tunnel* augmente le nombre de positions possibles pour l'objet, et donc la précision. De plus, agrandir un *tunnel* permet d'accéder plus facilement aux différentes valeurs d'une échelle discrète, puisqu'elles sont plus larges. A contrario, les modulations rapides mais peu précises peuvent être effectuées plus rapidement en réduisant la longueur du *tunnel*.

L'étirement se réalise avec deux mains, en saisissant chaque extrémité avec un rayon de Piivert et en écartant/resserrant les mains, comme on peut le voir sur la figure 4.19.

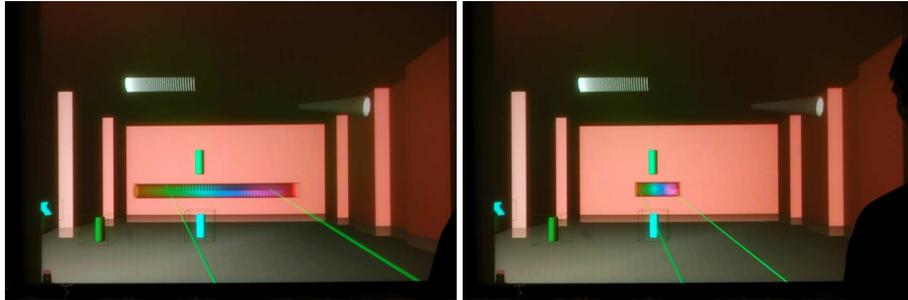


FIG. 4.19 – Les *tunnels* peuvent être étirés afin d’augmenter leur résolution.

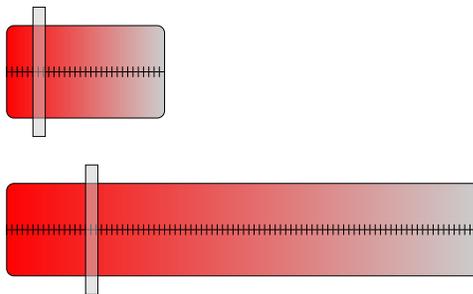


FIG. 4.20 – Précision des *tunnels*. Les objets du haut et du bas ont la même valeur, définie par leur position horizontale. En bas le *tunnel* étiré permet d’accéder à un plus grand nombre de valeurs du paramètre.

### Combinaison de paramètres

Plusieurs paramètres peuvent être associés à un seul *tunnel*, comme présenté sur la figure 4.21. On obtient ainsi des mappings geste-son de type "one-to-many", le déplacement horizontal permettant de modifier plusieurs paramètres sonores simultanément selon différentes échelles. Cependant, s’il y a séparation perceptive des paramètres graphiques, comme expliqué dans le chapitre 3, les mappings audiovisuels restent de type one-to-one, ce qui permet de visualiser les paramètres sonores modifiés séparément.

### Organisation spatiale

Les *tunnels*, tout comme les widgets réactifs 3D, peuvent être agencés librement dans l’environnement virtuel, comme un batteur positionnerait les éléments de sa batterie. Ainsi la configuration de *tunnels* peut évoluer au cours d’une performance, le musicien définissant leurs positions/tailles/accès, suivant les modulations qu’il souhaite effectuer dans différentes parties.

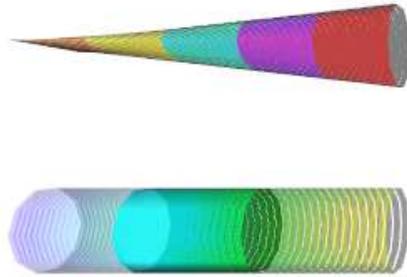


FIG. 4.21 – Combinaison de 2 et 3 paramètres sur des *tunnels*. En haut, Taille (continu) et teinte (discret). En bas, Transparence (discret), teinte (continu) et espacement (discret)

### 4.2.3 Perspectives

Les *tunnels* permettent donc d'effectuer les gestes de modulation graphique discrets et continus. Ils offrent une précision réglable, des combinaisons de paramètres contenant chacun plusieurs échelles, et peuvent être utilisés aussi bien pour la modification d'un seul ou de plusieurs objets 3D.

Cependant, les *tunnels* pourraient être améliorés sur plusieurs points.

- L'axe horizontal empêche d'utiliser le paramètre de largeur (étirement sur l'axe X) Il faudrait alors pouvoir orienter le *tunnel* selon l'axe Y afin de prendre en compte ce paramètre.
- Pour l'instant les paramètres sont combinés sur une seule dimension (axe horizontal). Il pourrait être intéressant d'avoir des *tunnels* à deux ou à trois dimensions, bien que cela poserait le problème de l'activation et du saut à une valeur précise des paramètres.
- Les *tunnels* à une dimension pourraient être définis non plus comme des cylindres, mais comme des courbes dans l'espace, ce qui permettrait d'avoir des "chemins 3D" de modulation prédéfinis, qui peuvent être intéressants musicalement.



## Chapitre 5

# Drile : un environnement virtuel immersif pour le live-looping hiérarchique

*Reality is that which, when you stop believing in it, doesn't go away.*  
Philip k. Dick

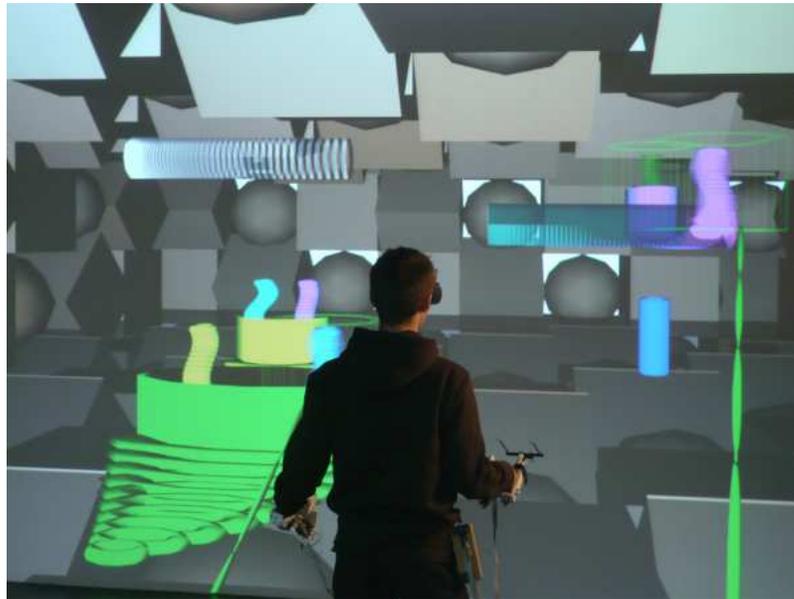
Dans ce chapitre, nous présentons *Drile*, un instrument multi-processus basé sur le live-looping hiérarchique, présenté dans le chapitre 2, en nous appuyant sur les widgets réactifs 3D et sur les techniques d'interaction musicales pour les environnements virtuels décrits dans les chapitres précédents. Nous décrivons ensuite deux installations matérielles qui tiennent compte des problématiques liées à la mise en œuvre de performances musicales.

### 5.1 Drile

*Drile* [11] est un instrument 3D immersif qui permet d'utiliser la technique du *live-looping hiérarchique*, c'est à dire de construire et de manipuler des arbres de live-looping. Cet instrument s'appuie sur le concept des *widgets réactifs 3D*, sur l'utilisation de *Piivert* et des *tunnels*. Un musicien l'utilisant est représenté sur la figure 5.1. Une vidéo de l'instrument est visible sur <http://vimeo.com/9206485>.

#### 5.1.1 Mise en œuvre

*Drile* est composé de trois applications tournant sur système d'exploitation Gnu/Linux : *drile-audio*, *drile-ui* et *drile-ui-display*. Elles sont lancées sur des machines différentes, notamment car le rendu graphique ou audio peut consommer beaucoup de ressources suivant le nombre de nœuds/arbres utilisés.

FIG. 5.1 – Un musicien utilisant *Drile*

Drile-audio gère les processus musicaux, donc le contenu des nœuds (séquences et audio), mais également les événements envoyés par Piivert. Elle s'appuie sur ALSA<sup>1</sup> pour Piivert, sur le serveur de son JACK<sup>2</sup> pour le rendu audio stéréo, sur plusieurs plug-ins LV2<sup>3</sup> pour les effets audio des nœuds, sur plusieurs plug-ins VAMP<sup>4</sup> pour l'analyse audio qui va servir à la visualisation des nœuds, et finalement sur la bibliothèque osc-Pack<sup>5</sup> pour la communication avec l'application drile-ui par le protocole OpenSound-Control [136]. La diffusion est effectuée grâce à une carte son externe et des enceintes actives.

Drile-ui gère le rendu de l'environnement 3D. Cette application s'appuie sur la bibliothèque de graphe de scène OpenSG. Celle-ci permet de distribuer l'affichage de la scène à plusieurs instances de l'application drile-ui-display. Dans notre cas, deux instances sont utilisées, sur deux machines différentes, afin de générer une image stéréoscopique en combinant deux points de vue, comme nous l'avons expliqué dans la section 1.3.3. Le musicien est équipé de lunettes stéréoscopiques passives Infitec et leur tête est suivie par le système de caméras infrarouges DTrack.

La configuration des applications (contenu des feuilles, effets audio, organisation des scènes) s'effectue dans des fichiers textes avec un syntaxe XML. Des exemples sont donnés en annexe de ce mémoire.

<sup>1</sup><http://www.alsa-project.org>

<sup>2</sup><http://jackaudio.org/>

<sup>3</sup><http://lv2plug.in/>

<sup>4</sup><http://www.vamp-plugins.org/>

<sup>5</sup><http://www.audiomulch.com/rossb/code/oscpack/>

L'organisation des différents composants de *Drile* est représentée sur la figure 5.2.

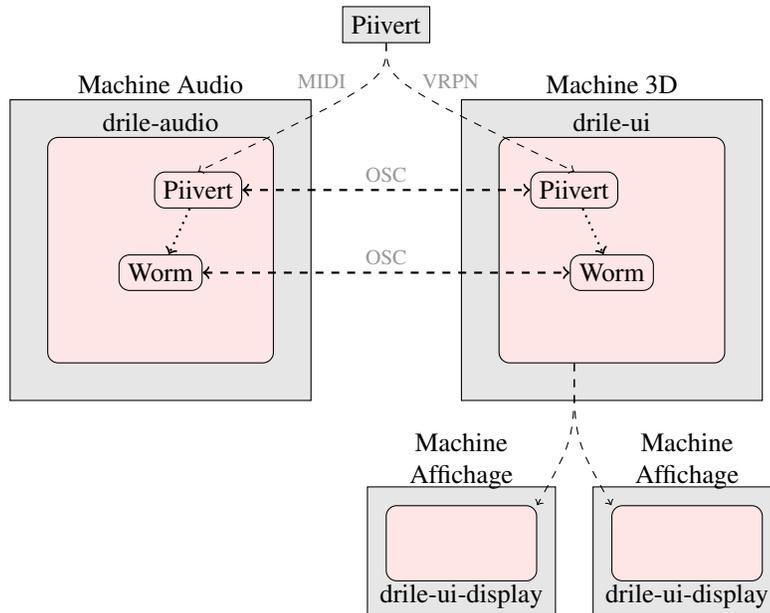


FIG. 5.2 – Composants matériels (en gris) et logiciels (en rose) de *Drile*. Les messages OpenSoundControl(OSC) et VRPN sont acheminés par Ethernet.

## 5.1.2 Hiérarchie

### Worms

Dans *Drile*, les nœuds des *arbres de live-looping* sont représentés par des widgets réactifs 3D appelés *worms*. Ces *worms* permettent de manipuler les paramètres des nœuds grâce aux mappings de contrôle suivants : Taille/Amplitude, Teinte et Luminance / Pitch, Transparence/Saturation, Dispersion/Réverbération. Des mappings de visualisation sont aussi définis : la forme d'un *worm* (16 anneaux superposés) reflète le spectre du flux audio sortant du nœud, la rotation autour de l'axe Y (vertical) donne la position de la tête de lecture pour les séquences correspondant au nœud (la forme est modifiée pour percevoir cette rotation). Ces mappings ont été choisis d'après les conclusions du chapitre 3, afin que les différents paramètres graphiques ne se perturbent pas mutuellement.

Le contenu musical des *worms* peut être déclenché avec *Piivert* en utilisant des gestes de bas niveau effectués avec les index, majeurs et annulaires, ce qui va correspondre aux gestes d'excitation présentés dans le chapitre 4. Les gestes "Hit" vont déclencher la lecture complète du contenu (séquence ou son), tandis que les gestes "Pressure" vont contrôler une synthèse granulaire mais uniquement sur les feuilles. Les

paramètres graphiques, et donc les effets des nœuds, peuvent être modifiés en utilisant les *tunnels*, ce qui va correspondre aux gestes de modulation.

### Plan d'interaction

*Piivert* est donc utilisé pour manipuler les *worms*. Comme nous l'avons décrit dans le chapitre 4, la sélection par rayon virtuel sur laquelle s'appuie *Piivert* est plus efficace à courte distance. De même, l'utilisation des *tunnels* pour la modulation graphique est facilitée si les tous les éléments graphiques se trouvent sur le même plan. Finalement, la représentation des *arbres de live-looping* ne doit pas perturber l'interaction. Il faut donc éviter que la structure 3D utilisée pour ces arbres surcharge l'espace d'interaction et empêche de sélectionner et de saisir les *tunnels* et les *worms*.

Nous proposons donc de maintenir les *tunnels* et les *worms* que l'on veut modifier sur un *plan d'interaction* orthogonal à l'axe de la caméra, et de représenter les arbres en organisant les *worms* sur la profondeur (selon l'axe de la caméra). Cette organisation est présentée sur la figure 5.3. Elle permet donc de visualiser et de manipuler correctement les arbres, tout en préservant une interaction efficace pour les opérations sur les nœuds car le plan d'interaction ne comporte que les éléments que l'on souhaite manipuler.

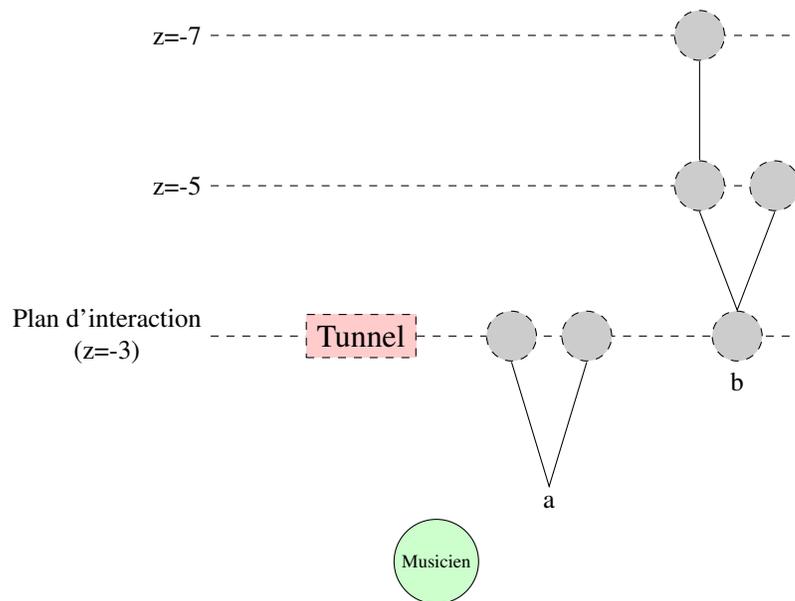


FIG. 5.3 – Plan d'interaction et arbres de live-looping alignés par niveaux. Les feuilles de l'arbre "a" sont accessibles car positionnées sur le plan d'interaction, leur parent n'est visible que par son appendice. L'arbre "b" est visible en entier. Seule la racine et ses deux enfants peuvent être saisis et passés dans le *tunnel* situé sur le *plan d'interaction*.

### Arbres

Les *arbres de live-looping* sont ainsi représentés par des *worms* connectés entre eux et organisés sur l'axe Z. Chaque niveau de l'arbre est placé à une distance définie, comme représenté sur la figure 5.3. Le niveau le plus près de l'utilisateur est placé sur le *plan d'interaction*. Chaque nœud parent possède un appendice dans lequel sont positionnés les nœuds enfants.

Toujours afin d'optimiser l'interaction, seuls les nœuds situés sur le *plan d'interaction* et leurs enfants directs peuvent être saisis et manipulés. L'accès à ces enfants est facilité par l'utilisation du suivi des mouvements de la tête de l'utilisateur, celui-ci peut en effet se décaler pour voir les nœuds situés derrière d'autres. Les enfants attrapés "sautent" alors au *plan d'interaction* afin de pouvoir être passés dans les *tunnels*. Lorsque un *worm* est déplacé sur le *plan d'interaction*, ses enfants restent derrière lui. S'il a un parent, il retrouve sa position dans l'appendice lorsqu'il est relâché.

Pour pouvoir manipuler les nœuds des autres niveaux, il est donc nécessaire de les amener au *plan d'interaction*. Ceci s'effectue en tirant/poussant un *worm* après l'avoir saisi. Tous les *worms* sont alors décalés d'un niveau. Lorsque l'on descend dans l'arbre, les *worms* se rapprochent donc tous du musicien, comme le montre la figure 5.4. Le *worm* situé sur le *plan d'interaction* disparaît alors, ne laissant que son appendice, afin de ne pas gêner la manipulation des enfants.

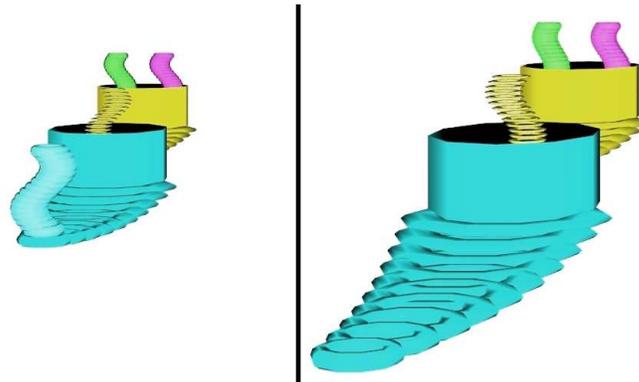


FIG. 5.4 – Arbre de live-looping à trois niveaux (à gauche). Saisir et tirer un *worm* vers soi descend d'un niveau dans l'arbre (à droite).

### Opérations sur les arbres

Les opérations effectuées sur les *arbres de live-looping* dans *Drile* sont représentées sur la figure 5.5.

L'opération de construction se doit d'être rapide et intégrée aux manipulations des *worms* puisqu'elle va conditionner l'enregistrement des séquences de ces manipulations (déclenchements/excitation et modifications des effets/modulation). Des gestes de haut niveau effectués avec Piivert sont donc utilisés pour cette opération, et l'opération

d'extraction. La notation de ces gestes est donnée sur la figure 5.6. La première étape est déclenchée en sélectionnant un *worm* avec *Piivert* et en effectuant un "Flam". Le *worm* se déforme alors et la séquence des événements est enregistrée. Un nouveau "Flam" déclenche la deuxième étape de l'opération. Un *worm* parent est créé avec un appendice, et le premier *worm* est reculé d'un niveau pour se retrouver dans l'appendice. La séquence enregistrée est lue en boucle et les rotations des deux *worms* reflètent la position de la tête de lecture.

L'opération fusion, moins critique temporellement, est déclenchée lorsque deux *worms* restent en contact pour plus d'une seconde. Un des parents est alors supprimé et les enfants sont regroupés dans l'appendice de l'autre. Leur position dans l'appendice n'a pour l'instant pas de signification particulière, mais il pourrait permettre de définir la séquence avec laquelle les autres sont synchronisées, plutôt que de choisir automatiquement la séquence la plus ancienne. La rotation du *worm* parent reflète alors la lecture de cette séquence "pulsation", tandis que les rotations des *worms* enfants reflètent les lectures de leurs séquences.

Comme expliqué dans la section 2.2.2, les opérations de construction et de fusion peuvent être combinées. Pour cela nous utilisons un paramètre du geste de haut niveau effectué avec *Piivert* pour la construction. Si deux *worms* sont sélectionnés avec les rayons dirigés par chaque main, et que le musicien effectue le même "Flam" mais avec un doigt de chaque main, comme le montre la figure 5.6, les parents créés pour chaque *worm* sont directement fusionnés. De plus, si la technique du "Split Ray", décrite dans le chapitre 4, est utilisée, ceci permet d'enregistrer simultanément les séquences de jusqu'à 6 *worms*.

L'opération de duplication n'est pas non plus critique temporellement, elle s'effectue donc en attrapant un *worm* avec deux rayons et en l'étirant. Ce *worm* et ses enfants sont alors dupliqués et il est amené au plan d'interaction s'il ne l'était pas déjà.

L'opération d'extraction est effectuée avec le même geste de haut niveau que l'opération de construction, mais en utilisant le paramètre de direction. L'ordre des doigts pour le "Flam" est donc inversé, comme on peut le voir sur la figure 5.6. Le *worm* est alors supprimé de son parent, extrait de l'arbre avec ses enfants et ramené au niveau du plan d'interaction. Si l'opération est appliquée sur un *worm* sans parent, comme expliqué dans la section 2.2.2, la lecture de la séquence est arrêtée et ce *worm* reprend sa forme initiale.

### 5.1.3 Pièces

Comme expliqué dans la section 2.2.2, des scènes de live-looping existent également pour le *live-looping hiérarchique*. Cependant, tandis que les scènes, numérotées, sont sélectionnées grâce à un potentiomètre sur les contrôleurs matériels et que des menus sont utilisées pour parcourir les noms de ces scènes dans les logiciels de live-looping 2D, les scènes dans *Drile* sont représentées par des pièces 3D.

Ces pièces sont organisées en grille et contiennent chacune un ensemble *d'arbres de live-looping* et un ensemble de tunnels. Les murs de ces pièces sont constitués de motifs d'objets 3D (primitives ou modèles) dont les paramètres (couleur, orientation, transparence, taille) peuvent être définis dans les fichiers de configuration. Ceci permet de décrire le contenu musical (sons des feuilles et séquences), c'est à dire l'ambiance de

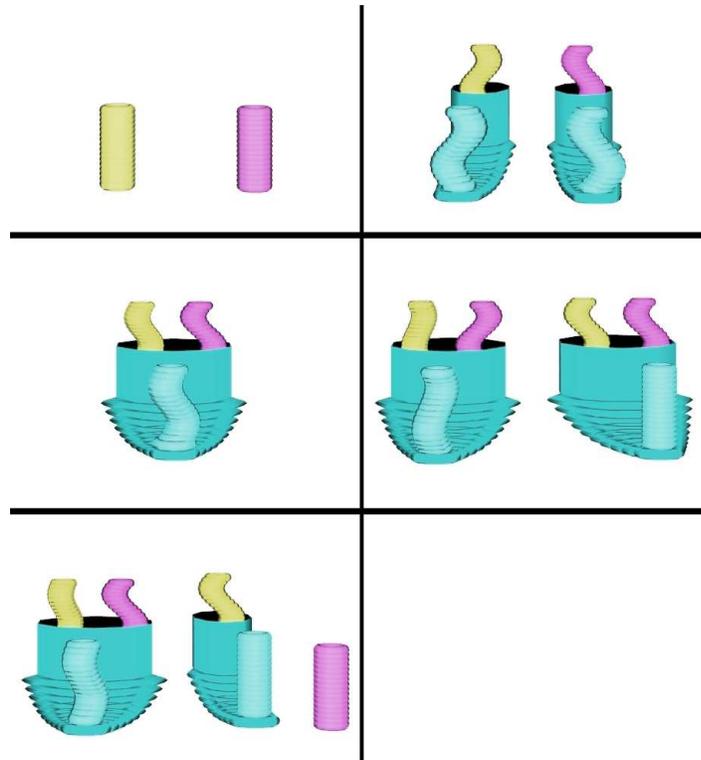


FIG. 5.5 – Opérations sur les arbres de live-looping (de gauche à droite, haut en bas) : Construction, Fusion, Duplication, Extraction

chaque pièce, visuellement. Ainsi une pièce dont les murs sont des sphères de couleurs vives correspondra à une ambiance plutôt "joyeuse" mais "calme", tandis qu'une pièce sombre dont les murs sont hérissés d'épines renverra évidemment à une toute autre ambiance. En plus des sons différents, l'organisation sous forme de pièces nous permet de sélectionner des *tunnels* différents et ainsi de limiter les manipulations musicales suivant l'ambiance.

Dans le cadre d'une performance, il est également possible de déplacer un *worm* ou un arbre complet d'une scène vers une autre. Ceci permet d'effectuer des transitions entre différentes parties, en partant d'un arbre précédemment construit et en y ajoutant des séquences d'une nouvelle scène, ou en coupant la lecture d'un arbre tout en commençant une nouvelle partie musicale.

Les pièces contiennent également un sol translucide. Lorsqu'un *worm* ou arbre est déplacé sous ce sol, son flux audio est redirigé sur une autre sortie stéréo de drile-audio, ce qui peut permettre de couper facilement le volume d'un arbre, mais également de préparer des boucles uniquement au casque, comme le ferait un DJ, avant de les intégrer à la scène et donc au rendu principal.

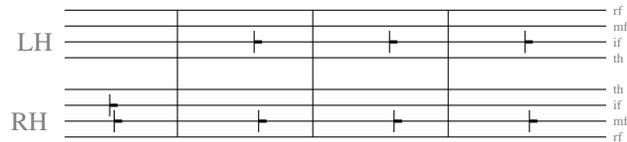


FIG. 5.6 – Gestes *Pivert* pour les opérations de construction et d'extraction à une main puis de construction à deux mains (construction+fusion).

Le déplacement d'une pièce à une autre se fait en plusieurs étapes, et s'appuie sur le déplacement physique du musicien grâce au suivi des mouvements de sa tête. Quand le musicien recule au delà d'un certain seuil, la caméra virtuelle, jusqu'alors située à l'intérieur d'une scène, recule en révélant une grille de scènes, comme présenté sur la figure 5.7. Le musicien peut alors sélectionner une des scènes de la grille simplement en orientant sa tête dans la direction de cette scène. Lorsqu'il avance à nouveau vers l'écran, la caméra se déplace automatiquement jusqu'à la scène choisie.

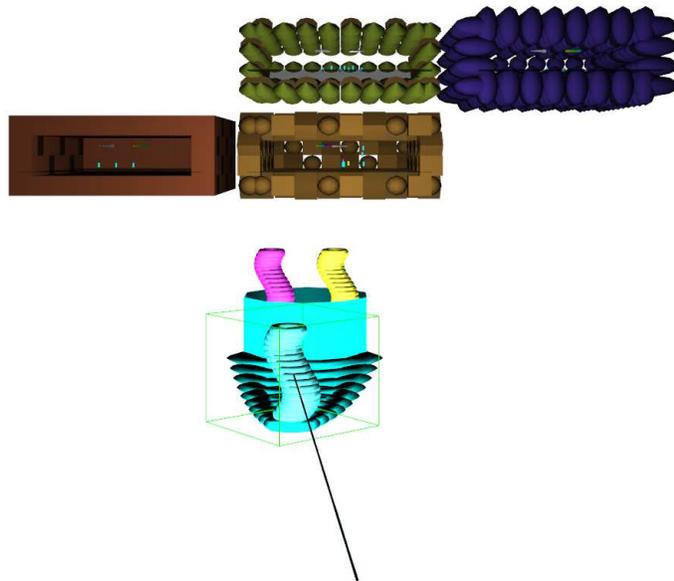


FIG. 5.7 – Vue de plusieurs scènes. Dans ce cas, le musicien a attrapé un arbre de live-looping et le déplace d'une scène à une autre.

### 5.1.4 Apprentissage

Le *live-looping hiérarchique* est adapté à une utilisation par des musiciens de niveaux différents grâce à sa structure arborescente. Ceci est accentué dans *Drile*. En effet, une technique d'interaction supplémentaire a été développée afin de réduire encore l'expertise nécessaire aux utilisateurs débutants : le *bucket*. Il s'agit d'un "panier" virtuel qui peut être manipulé à l'aide d'un dispositif simple (suivi de position et d'orientation et un seul bouton poussoir). Lors d'une première utilisation de *Drile* les musiciens se servent du *bucket* et ne peuvent saisir que les *worms* qui se trouvent sur le plan d'interaction et les passer au travers des *tunnels*, car le *bucket* est bloqué sur le plan d'interaction. Ainsi leur seule action possible sur la musique est la modification des effets sur des arbres pré-enregistrés. Suivant les *tunnels* accessibles, les utilisateurs débutants affecteront donc plus ou moins la musicalité des séquences. Ils pourront également couper le volume des séquences en plaçant les arbres sous le sol de la scène.

Après s'être habitué à l'environnement et à la manipulation des *worms*, les utilisateurs novices peuvent commencer à utiliser *Piivert*. Comme nous l'expliquions dans la section 2.2.2, ils peuvent facilement jouer des séquences musicales intéressantes en n'interagissant qu'avec les *worms* racines. A mesure qu'ils acquièrent de l'expérience, ils peuvent descendre dans les *arbres de live-looping* afin de manipuler les *worms* ayant un contenu plus brut et donc nécessitant plus d'interaction. En arrivant aux *worms* feuilles, ils peuvent directement jouer les sons élémentaires, ce qui demande une certaine pratique musicale afin d'obtenir des résultats intéressants et de pouvoir s'intégrer aux séquences de plus haut niveau. Finalement, les utilisateurs confirmés pourront effectuer les opérations sur les arbres, et ainsi construire et modifier ces derniers.

Si l'on analyse *Drile* selon les "Dimension Space" présentés dans le chapitre 1, il semble se situer entre deux configurations, comme on peut le voir sur la figure 5.8. En effet, tandis que les débutants utilisant le *bucket* ont un niveau de contrôle musical et des degrés de liberté restreints, ces derniers augmentent avec l'utilisation de *Piivert* et avec la descente dans les *arbres de live-looping*. Parallèlement l'expertise requise augmente.

### 5.1.5 Collaboration

*Drile* favorise également la collaboration, aussi bien entre musiciens de niveaux différents, qu'entre musiciens experts.

Ainsi, un musicien expert utilisant *Piivert* peut jouer avec plusieurs musiciens débutants équipés de *bucket* sur un même écran, comme le présente la figure 5.9. Alors que l'utilisation de *Piivert* nécessite un suivi des mouvements de la tête afin de pouvoir correctement percevoir la position et l'orientation des rayons virtuels, les *buckets* sont plus simples, ils ne se déplacent que sur un plan 2D et donc le suivi de tête n'est pas nécessaire. Le musicien expert peut construire les *arbres de live-looping* en enregistrant les séquences, afin d'obtenir un résultat musicalement intéressant, et ensuite les faire passer aux novices, qui n'auront qu'un contrôle limité. De même, l'expert peut sélectionner les *tunnels* que les novices vont pouvoir utiliser, et ainsi limiter l'impact qu'ils auront sur le rendu musical.

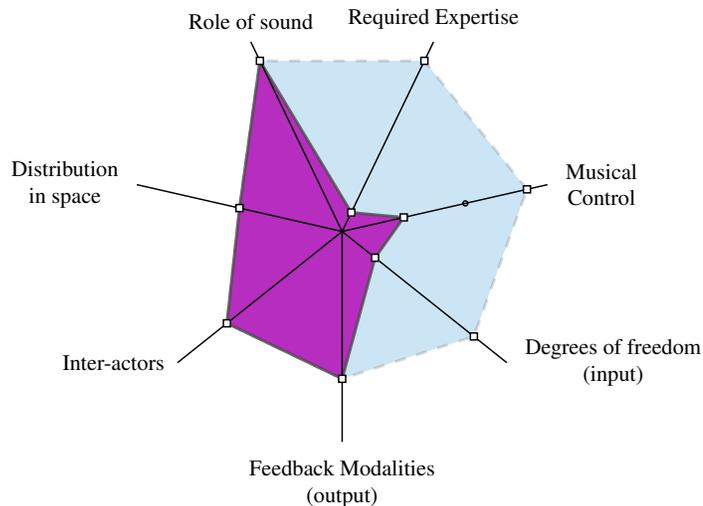


FIG. 5.8 – "Dimension Space" de *Drile*, en violet pour les novices, et bleu pour les utilisateurs experts.

Plusieurs experts équipés de *Piivert* peuvent également collaborer devant un même écran. Cela nécessite cependant le suivi des mouvements des têtes de chaque musicien, afin qu'ils aient chacun une vue de l'environnement adaptée à leur position, comme nous l'avons expliqué dans la section 1.3.3. Des solutions s'appuyant sur plusieurs écrans, et même des interactions à distance, pourraient également être mises en place. Les musiciens peuvent alors, comme nous l'avons décrit dans la section 2.2.2, travailler ensemble de plusieurs façons. Certains vont ainsi enregistrer des séquences et les fusionner les nouveaux *worms* à un arbre commun tandis que d'autres vont se concentrer sur la manipulation de cet arbre et de ses *worms*. Des *arbres de live-looping* peuvent également être dupliqués afin que chacun développe sa version avec des variations particulières et la joue simultanément.

### 5.1.6 Perspectives

*Drile* met en œuvre la technique du *live-looping hiérarchique* de manière efficace. Comme nous l'énonçons dans le chapitre 2, contrairement aux interfaces 2D, l'environnement virtuel facilite la mise en œuvre de cet instrument multi-processus. En effet, la profondeur nous permet de visualiser les arbres sans surcharger et donc perturber le *plan d'interaction*. De même, des contrôleurs matériels ne donneraient pas accès à tous les éléments de structures dynamiques.

Les musiciens peuvent manipuler efficacement les nœuds représentés par les *worms*, en utilisant *Piivert*, les *tunnels* et le *plan d'interaction*. De même les opérations sur les arbres sont correctement intégrées à l'interaction musicale grâce aux gestes de haut niveau effectués avec *Piivert*. Finalement, les pièces 3D sont adaptées pour donner



FIG. 5.9 – Collaboration entre un utilisateur avancé utilisant Piivert (à gauche) et deux utilisateurs débutants utilisant les *buckets*

des informations sur le contenu musical des scènes de live-looping.

Plusieurs pistes seraient intéressantes à suivre pour le développement de *Drile*.

- Notre implémentation s’appuie sur des fichiers sons multi-échantillonnés comme contenu des nœuds. D’autres types de synthèse pourraient cependant être utilisés. Chaque scène pourrait correspondre à un type de synthèse ou ceux-ci pourraient être mélangés.
- Ces scènes 3D pourraient être placées sur un chemin plutôt que grille afin de forcer la progression dans une structure musicale prédéfinie, sans pour autant influencer la musique jouée dans chaque pièce.
- Finalement, *Drile* pourrait permettre de pousser le live-looping à un niveau supérieur, en enregistrant non plus seulement les modifications des nœuds et des des arbres, mais également les actions des musiciens. Chaque enregistrement produirait un avatar 3D du musicien, répétant la série d’actions effectuées. Les musiciens pourraient alors jouer avec ces avatars, voire les diriger comme le ferait un chef d’orchestre.

Finalement, le *live-looping hiérarchique* et particulièrement *Drile* étant destinés à la performance musicale, il est important de réfléchir aux problématiques liées à l’utilisation d’un instrument virtuel immersif, aussi bien pour les musiciens que pour le public. La section 5.2 donne plusieurs pistes pour la mise en œuvre de représentations publiques s’appuyant sur des environnements musicaux immersif.

## 5.2 Installations pour la performance musicale immersive

### 5.2.1 Performances musicales immersives

Contrairement aux concerts d'instruments traditionnels, les performances de musique électronique, s'appuyant sur des contrôleurs matériels alternatifs ou simplement sur des ordinateurs portables, peuvent perturber les spectateurs, et ceci pour plusieurs raisons.

#### Compréhension des instruments

La première est la difficulté de compréhension des nouveaux instruments électroniques. Les modes de fonctionnement des instruments acoustiques, de par leurs catégories (instrument à vents, à cordes, percussions...) et leurs propriétés physiques (matériaux, tailles, ...), sont connus ou relativement évidents. Ils sont de plus figés, il n'évoluent pas au cours d'une performance. D'autres instruments sont suffisamment courants pour que leur fonctionnement soit connu par les spectateurs.

Les nouveaux instruments sont par contre souvent utilisés uniquement par leurs développeurs, ou du moins par un petit groupe de musiciens. Ils n'ont quasiment aucune contraintes en terme de processus de synthèse sonore, de mappings et de connexions. Il est donc impossible à priori pour les spectateurs de connaître leur fonctionnement à partir du résultat sonore. De plus, ce fonctionnement peut varier au cours du temps, en fonction des différentes parties d'une performance par exemple.

L'utilisation d'interfaces graphiques permet d'améliorer cette compréhension, si elles donnent suffisamment d'informations sur les processus sonores, leurs paramètres et leur organisation (et ne sont pas seulement des ensembles de potentiomètres). Les *widgets réactifs 3D* sont particulièrement adaptés à cette visualisation de processus sonores, comme nous l'expliquons dans le chapitre 3.

#### Compréhension des actions des musiciens

En plus de la compréhension du fonctionnement des instruments, la compréhension des actions des musiciens et de leur influence sur l'instrument est primordiale. Dans le cas des instruments multi-processus, comment en effet distinguer les variations programmées des variations réalisées par les musiciens ? Par exemple, comment comprendre l'action exacte d'un seul potentiomètre d'une surface de contrôle si un nombre important de paramètres sonores varient simultanément ?

Dans certains types de performances, tels que les concerts d'ordinateurs portables, le jeu du musicien est constitué sur de légers mouvements de doigts et de poignet (souvent cachés par un écran) et donc imperceptibles pour les spectateurs. Un exemple est présenté sur la figure 5.10. Comme le précise Cascone [31], le public peut alors se sentir frustré voir abusé par le musicien. Il ne sait pas comment fonctionne l'instrument, ni si le musicien a une influence sur la musique (ou s'il se contente de lire un enregistrement de sa performance tout en parcourant ses e-mails).



FIG. 5.10 – Performance de musique électronique avec ordinateur portable et surface de contrôle

Il est donc essentiel d'amplifier les gestes des musiciens et surtout leur incidence sur les instruments. Encore une fois les instruments s'appuyant sur des interfaces graphiques permettent de visualiser ces actions. Ces interfaces doivent cependant être visibles du public. Par exemple, lors des performances publiques de la Reactable [73], la surface de l'instrument est filmée et projetée afin que les spectateurs aient une vision de l'interface et des gestes des musiciens.

### **Communication Musiciens - Spectateurs**

Un autre élément essentiel est la possibilité pour le public et les musiciens de communiquer. En effet, comme l'indique Bongers [16], la performance inclue une communication, souvent non verbale, entre les spectateurs et les artistes. Cette communication, fondamentale, n'est évidemment possible que si les contacts sont les plus nombreux. Il faut donc éviter de supprimer ces contacts, qu'il soient visuels ou auditifs.

La mise en œuvre de performances s'appuyant sur des environnements virtuels immersifs doit tenir compte de ces conditions. Il est donc nécessaire de réfléchir à leur organisation matérielle (affichage, captations) et spatiale (position des spectateurs, musiciens).

### Compréhension de l'instrument et du jeu

Comme nous l'avons décrit dans le chapitre 1, l'utilisation d'environnements virtuels immersifs pour la performance musicale présente un avantage par rapport aux interfaces 2D. En effet, en immergeant les spectateurs dans l'environnement, l'intégration de l'instrument virtuel dans l'espace est améliorée. Mieux le public perçoit l'instrument, les différents éléments graphiques et les actions des musiciens, mieux il le comprend. De plus, dans le cas du rayon virtuel par exemple, il y a une continuité entre les dispositifs d'interaction réels et ce rayon. Les gestes du musicien sont donc amplifiés graphiquement, leur impact est évident.

Cependant ceci nécessite que le public ait une perception correcte de l'environnement virtuel ainsi que du musicien. Cela signifie notamment qu'il ne doit pas y avoir d'occultations de l'affichage et que le musicien doit être de préférence face au public. Ainsi, l'installation mise en place pour *Drile*, présentée sur la figure 5.11, dans laquelle le musicien est devant l'écran, est à éviter car le musicien cache une partie de l'environnement au public et qu'il est de dos. Il en serait de même dans le cas d'une CAVE. Il n'est pas non plus envisageable de placer le musicien derrière le public sans perturber la majorité des techniques d'interaction utilisées.

Les musiciens ont également besoin d'un système de suivi de tête afin d'interagir correctement, par exemple à l'aide de "rayons virtuels" dans le cas de *Piivert*, avec l'environnement en 3D. Cependant, si les spectateurs utilisent le même affichage que les musiciens, le changement de vue adapté au musicien ne va pas correspondre aux mouvements de têtes du public. Les mouvements de la tête du musicien vont donc perturber la perception des spectateurs, à moins d'utiliser un système de multi stéréo [50].

Une solution permettant d'éviter à la fois les occultations et les problèmes de modification de vue est d'utiliser des écrans différents pour le public et les musiciens. Il faut cependant s'assurer que les gestes des musiciens restent visibles et que la continuité avec l'environnement virtuel est préservée.

### Communication Musiciens - Spectateurs

L'immersion contraint la communication entre les musiciens et les spectateurs. Ainsi dans le cas présenté sur la figure 5.11, il n'y a aucun contact visuel possible. Le musicien ne voit pas le public, et le public voit le musicien seulement de dos.

Cette communication peut se faire virtuellement, auquel cas il faut intégrer les musiciens et le public dans l'environnement virtuel. Elle peut également se faire réellement en fonction de l'organisation spatiale de la performance.

#### 5.2.2 Scène Mixte

Cette première scène remplit les conditions données en se plaçant dans une approche mixte réel-virtuel.

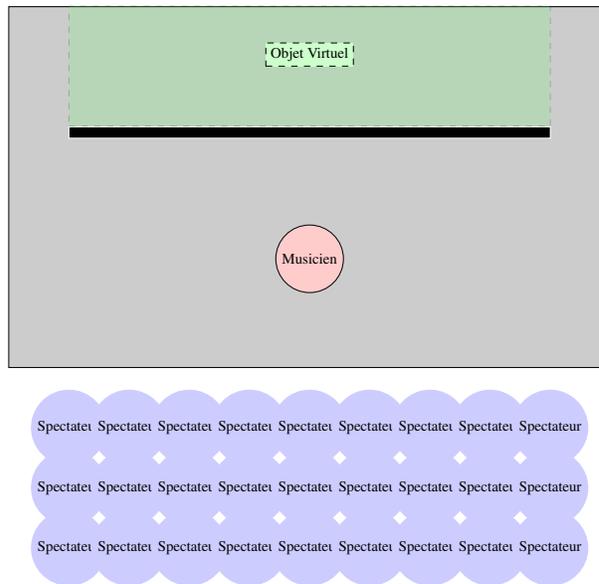


FIG. 5.11 – Dans une configuration simple, le musicien occulte une partie de l’environnement virtuel, et la vue est modifiée par le suivi de tête.

### Principe

S’il n’est pas possible, pour les raisons énoncées, d’avoir un écran commun pour le musicien et le public, il est néanmoins essentiel de préserver la cohérence de l’environnement virtuel par rapport aux actions du musicien, afin de ne pas affecter l’immersion. Une solution est donc de se rapprocher des systèmes d’affichage collaboratifs, tels que l’Illusion Hole [74] ou le Virtual Showcase [13]. Cependant ces systèmes sont essentiellement destinés à la visualisation et manipulation d’objets 3D relativement petits, tandis que les instruments 3D immersifs tels que *Drile* sont beaucoup plus volumineux.

La solution que nous proposons s’appuie sur un espace virtuel délimité par deux écrans, un pour les musiciens et l’autre pour les spectateurs. Elle est représentée sur la figure 5.12 et les vues respectives des musiciens et des spectateurs sont données sur la figure 5.13.

Les musiciens (ou du moins l’un d’entre eux) peut bénéficier du suivi de tête, son écran n’est pas partiellement caché par le public. Il peut de plus voir du public sur sa gauche en tournant plus ou moins la tête..

Les spectateurs voient les musiciens de côté pour les plus à droite, et de trois quart pour les plus à gauche. Ils perçoivent donc relativement bien leurs gestes. L’écran n’est pas occulté et il est orienté, tout comme le point de vue de l’environnement, de manière à ce que les objets virtuels soient vus de trois quart et que les rayons virtuels semblent sortir directement des dispositifs utilisés par les musiciens, dans ce cas *Pivert*. Allié

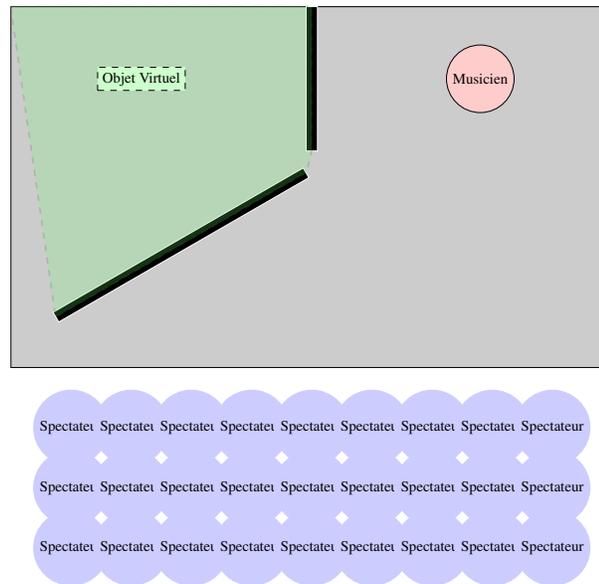


FIG. 5.12 – Scène Mixte

à l’affichage stéréoscopique, l’impression donnée aux spectateurs est que l’instrument virtuel se situe bien dans la pièce, à gauche des musiciens.

L’immersion est renforcée par l’ajout de diodes vertes aux extrémités de Piivert, représentant ainsi la base des rayons virtuels, et par la projection sur le musicien des couleurs ambiantes de l’environnement virtuel.

Cette *scène mixte* a ainsi deux composantes :

- Une partie réelle contenant les musiciens équipés de dispositifs d’interaction.
- Une partie virtuelle contenant l’instrument 3D immersif et intégrée à la scène réelle du point de vue des spectateurs.

### Mise en œuvre

Les composants matériels sont sensiblement les mêmes que pour *Drile*, décrits dans le chapitre 5. Trois ordinateurs sont ajoutés, lesquels sont lancées des instances de l’application *drile-ui-display*. Deux vont produire la vue stéréoscopique pour les spectateurs et le dernier va servir à projeter l’environnement sur les musiciens.

Les deux affichages sont en rétro projection et sont constitués chacun de deux vidéoprojecteurs utilisant la technologie INFITEC et d’écrans semi-transparents. Un vidéoprojecteur supplémentaire est dirigé vers les musiciens.

Les deux projections sont réalisées grâce à la bibliothèque OpenGL. Les coordonnées des deux écrans dans le repère de l’environnement virtuel, couplées aux positions des utilisateurs, permettent d’en calculer les matrices de projection. La position

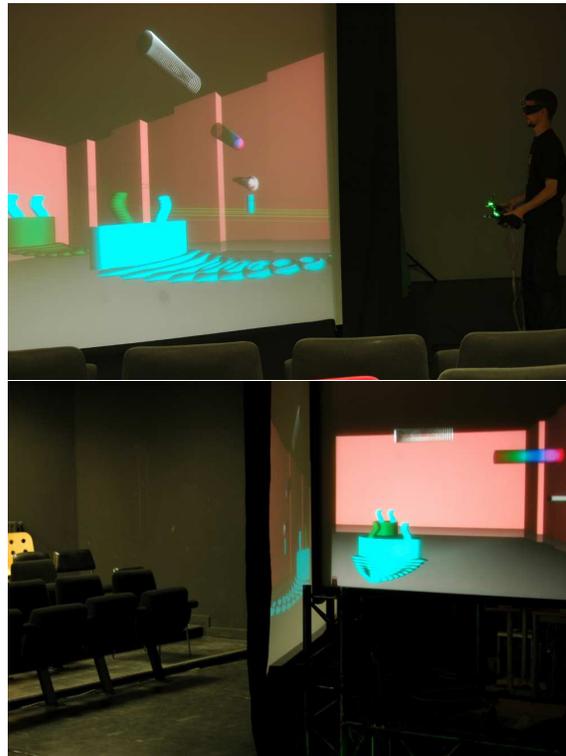


FIG. 5.13 – Scène Mixte : vues des spectateurs (en haut) et du musicien (en bas)

du musicien est ainsi captée grâce à un système de vision infrarouge, tandis que la position du public est définie empiriquement au centre de l'espace qui lui est réservé.

### Limitations

Bien que cette *scène mixte* satisfasse les contraintes d'immersion et de communication tout en étant relativement simple à mettre en œuvre, elle possède également certaines limitations.

- La vue de l'environnement est fixée à une position particulière au milieu du public. Le spectateur à cette position est parfaitement immergé. Le point de vue est cependant de moins en moins correct à mesure que l'on s'en écarte. Cependant l'immersion reste peu dégradée et la compréhension de la performance par le public n'en souffre pas.
- Un deuxième inconvénient de cette approche est que le point de vue du public est fixe. Le déplacer, par exemple afin de concentrer la vue sur un des éléments de l'environnement virtuel, rompt la continuité entre les parties réelles et virtuelles de la scène puisque les dispositifs réels (Piivert) et les éléments virtuels (rayons)

ne sont plus alignés.

- Finalement, cette *scène virtuelle* convient plutôt pour un nombre limité de spectateurs, afin de limiter l'écart avec le point de vue correct et la distance avec le musicien, qui va perturber l'immersion.

### 5.2.3 Scène Virtuelle

#### Principe

Le principe des écrans séparés est encore utilisé afin d'assurer une bonne immersion au public comme au musicien. Cependant, afin de remédier aux problèmes de déplacement dans l'environnement, les musiciens ne sont plus visibles réellement. Comme on peut le voir sur le schéma 5.14, ils peuvent être dissimulés derrière un drap, ou directement derrière l'écran. En supprimant ainsi les risques de discontinuités liés à la scène mixte, le public peut naviguer à l'intérieur de l'environnement. Un opérateur a la charge de déplacer la caméra virtuelle afin d'observer telle action plus précisément, ou simplement afin de changer d'angle de vue.

Pour assurer la communication, il est nécessaire que le public et les musiciens se voient mutuellement. Les musiciens sont donc reconstitués en 3D et intégrés à l'environnement afin que le public puisse percevoir tous leurs gestes. Cela peut être fait de manière non réaliste, avec un modèle 3D prédéfini animé selon le suivi de la tête et des mains d'un musicien, mais également de manière réaliste, en utilisant un système vidéo pour reconstruire les musiciens en 3D avec leurs textures réelles. Le public est également filmé mais ce flux vidéo est seulement appliqué comme texture à une surface, en face des musiciens. De cette manière, la communication est possible dans les deux sens, même si elle reste indirecte.

#### Mise en œuvre

Cette scène virtuelle n'a pas encore été testée, mais le système de reconstruction est fonctionnel.

La reconstruction 3D est effectuée à l'aide du système développé par la société 4DView issu des travaux du projet grImage [1]. Plusieurs caméras, situées autour du musicien, permettent d'obtenir plusieurs silhouettes et les textures associées. Le système définit alors le maillage 3D et l'envoie, avec les textures, à la machine sur laquelle est effectuée le rendu de l'environnement. Le maillage est alors ajouté comme nœud au graphe de scène géré par OpenSG.

Le public est filmé par une caméra située au-dessus de l'écran, le flux vidéo est acquis grâce à Video4Linux et affiché comme texture grâce à la fonctionnalité incluse dans OpenSG.

#### Limitations

Cette *scène virtuelle* permet au public de naviguer dans l'environnement et donc d'apprécier la performance sous plusieurs angles. Cependant elle est plus complexe

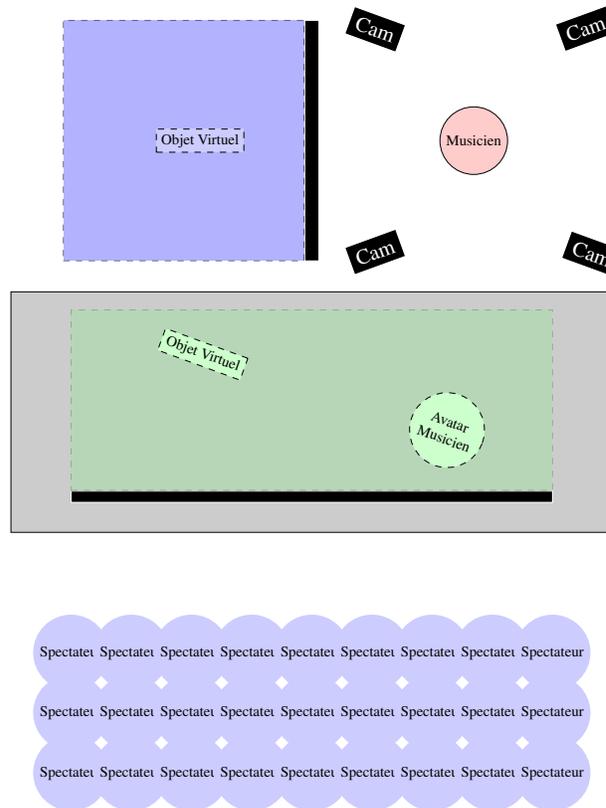


FIG. 5.14 – Scène Virtuelle : Le musicien réel n’est pas visible du public et interagit sur un écran séparé. Un avatar 3D, reconstitué grâce à des caméras, le représente dans la scène affichée pour le public.

à mettre en œuvre que la *scène mixte*, notamment si elle s’appuie sur la reconstruction 3D réaliste des musiciens, ce qui est évidemment préférable à l’utilisation d’avatars 3D modélisés. La communication indirecte (virtuelle), évidemment moins précise et intéressante que la communication réelle, peut également être perturbée par les problèmes de latence induit par le système de reconstruction.

#### 5.2.4 Perspectives

La première piste est évidemment la mise en œuvre de de cette scène virtuelle.

Le choix entre la *scène mixte* et la *scène virtuelle* dépend essentiellement du nombre de spectateurs. Pour un public restreint, la présence réelle des musiciens améliore la compréhension de leurs actions et la communication. Au-dessus d’un certain nombre de spectateurs cependant, un point de vue correct pour tous, et donc une

immersion suffisante, devient difficile à assurer. La *scène virtuelle* va alors permettre de naviguer dans l'environnement pour mettre en évidence les différentes actions des musiciens.

Il serait cependant intéressant de pouvoir basculer d'une configuration à une autre au cours d'une performance. Le système de reconstruction pourrait être ajouté à la *scène mixte* et le passage de l'un à l'autre s'effectuerait simplement en tirant un rideau afin de dissimuler les musiciens.

Finalement, comme le propose Bongers [16], l'interaction entre le public et le musicien pourrait être augmentée par le biais de l'instrument. Les mouvements du public pourrait entraîner des modifications de l'environnement virtuel, par exemple des changements de couleurs pour les pièces de *Drile*. Ceci permettrait d'amplifier la communication dans le sens public - musiciens.

# Conclusion

## Réalisations

Tout au long de ce mémoire nous avons cherché à adapter les dispositifs et techniques liées aux environnements virtuels immersifs aux spécificités de la performance musicale. Ceci nous a finalement permis de développer *Drile*, un instrument multi-processus de *live-looping hiérarchique* qui ne pouvait être mis en œuvre ni avec un contrôleur matériel ni avec une interface 2D, comme nous l'avons expliqué dans le chapitre 2.

Après avoir explicité les avantages des environnements virtuels et de l'interaction 3D pour les instruments multi-processus, et en particulier pour la technique du *live-looping hiérarchique* que nous avons mise au point, nous nous sommes penchés sur les problématiques de représentations et de manipulation des processus sonores dans ces environnements.

Nous avons ainsi proposé les *widgets réactifs 3D*. Ces composants graphiques permettent à la fois de visualiser et de manipuler simultanément plusieurs paramètres sonores grâce à des combinaisons de mappings audiovisuels s'inspirant de la perception humaine et que nous avons évalués.

Afin de manipuler ces *widgets réactifs 3D* de manière efficace, nous avons mis au point un dispositif d'interaction appelé *Piivert* et plusieurs techniques associées, dont les *tunnels*, en répartissant les gestes instrumentaux entre l'interaction graphique plus lente et des capteurs de pression plus précis.

Finalement, nous nous sommes appuyés sur ces techniques pour développer *Drile*, un environnement virtuel pour le *live-looping hiérarchique*. Nous avons également étudié comment adapter la mise en œuvre de *Drile* aux spécificités des performances publiques musicales.

## Perspectives

Parmi toutes les perspectives énoncées dans les chapitres 3 4 et 5, certaines nous semblent particulièrement intéressantes, notamment au niveau des mappings audiovisuels. Nous pensons en effet qu'il est essentiel de s'appuyer sur les travaux issus du domaine de la visualisation d'information afin d'optimiser la représentation graphique des paramètres sonores. Des tests utilisateurs supplémentaires pourraient permettre d'évaluer pour chaque paramètre graphique les différentes échelles liées à la

perception dans le cadre de l'interaction musicale.

Les techniques d'interaction pour *Piivert*, en particulier les gestes de percussion de haut niveau, pourraient être étendues, comme nous l'avons expliqué dans le chapitre 4, et également intégrées à des applications non-musicales mais nécessitant néanmoins une grande précision temporelle et une grande expressivité. Les *tunnels* pourraient évoluer vers la prise en compte de plusieurs dimensions pour les paramètres, et également permettre des modifications plus fines de la forme 3D.

Finalement, les possibilités de construction musicale collaborative offertes par le *live-looping hiérarchique* sont relativement peu exploitées dans *Drile*. De nouvelles techniques d'interaction pourraient être développées pour la collaboration non seulement locale mais également distante grâce à l'intégration de musiciens réels ou virtuels dans l'environnement virtuel. On peut imaginer la création de clones virtuels des musiciens, reproduisant certains gestes selon la technique du live-looping, que les musiciens réels pourraient diriger, ajoutant ainsi un niveau de contrôle au-dessus des arbres de live-looping.

Cette intégration de musiciens virtuels nécessite la mise en oeuvre de la *scène virtuelle*, utilisant les travaux du projet Grimage [1], et qui, comme nous le proposons dans le chapitre 5, pourrait se combiner avec la *scène mixte*.

### ***Jouer de Drile***

Finalement, *Drile* doit devenir un instrument à part entière. Ceci nécessite tout d'abord de permettre l'utilisation de dispositifs d'interaction autres que *Piivert* et de simplifier la création et la configuration des scènes par une interface graphique, afin de le rendre accessible au plus grand nombre de musiciens. Il faut également réfléchir aux moyens techniques (projection, tracking) et logistiques appropriés pour pouvoir facilement adapter *Drile* aux conditions des concerts traditionnels.

*Drile* ne doit pas rester un instrument de laboratoire !

# Bibliographie

- [1] J. Allard, J.-S. Franco, C. Menier, E. Boyer, and B. Raffin. The grimage platform : A mixed reality environment for interactions. In *ICVS '06 : Proceedings of the Fourth IEEE International Conference on Computer Vision Systems*, page 46, Washington, DC, USA, 2006. IEEE Computer Society.
- [2] D. Arfib, J.-J. Filatriau, and J.-M. Couturier. Using visual textures for sonic textures production and control. In *Proceedings of the 9th International Conference on Digital Audio Effects (DAFX-06)*, 2006.
- [3] B. Arnaldi, P. Fuchs, and J. Tisseau. *Chapitre 1*. Les Presses de l'École des Mines de Paris, 2003.
- [4] F. Attneave and R. K. Olson. Pitch as a medium : A new approach to psychophysical scaling. *The American Journal of Psychology*, Vol. 84, No. 2, pages 147–166, 1971.
- [5] S. P. Bacon and N. F. Viemeister. Temporal modulation transfer functions in normal-hearing and hearing-impaired listeners. *International Journal of Audiology*, 24(2) :117–134, 1985.
- [6] T. Barri. Versum : audiovisual composing in 3d. Copenhagen, Denmark, 18--21 May 2009. Re :New – Digital Arts Forum, Re :New – Digital Arts Forum.
- [7] M. Beaudoin-Lafon. *Interfaces homme-machine et création musicale*, chapter Moins d'interface pour plus d'interaction. Hermès Sciences, 1999.
- [8] A. J. Berkhout, D. de Vries, and P. Vogel. Acoustic control by wave field synthesis. *The Journal of the Acoustical Society of America*, 93(5) :2764–2778, 1993.
- [9] F. Berthaut, M. Desainte-Catherine, and M. Hachet. Interaction with the 3d reactive widgets for musical performance. In *Proceedings of Brazilian Symposium on Computer Music (SBCM09)*, 2009.
- [10] F. Berthaut, M. Desainte-Catherine, and M. Hachet. Combining audiovisual mappings for 3d musical interaction. In *Proceedings of the International Computer Music Conference (ICMC10)*, 2010.
- [11] F. Berthaut, M. Desainte-Catherine, and M. Hachet. Drile : an immersive environment for hierarchical live-looping. In *Proceedings of New Interfaces for Musical Expression (NIME10)*, 2010.

- [12] F. Berthaut, M. Hachet, and M. Desainte-Catherine. Piivert : Percussion-based interaction for immersive virtual environments. In *Proceedings of the IEEE Symposium on 3D User Interfaces*, 2010.
- [13] O. Bimber, B. Fröhlich, D. Schmalstieg, and L. M. E. ao. The virtual showcase. *IEEE Computer Graphics and Applications*, 21 :48–55, 2001.
- [14] D. Birnbaum, R. Fiebrink, J. Malloch, and M. M. Wanderley. Towards a dimension space for musical devices. In *NIME '05 : Proceedings of the 2005 conference on New interfaces for musical expression*, pages 192–195, Singapore, 2005. National University of Singapore.
- [15] T. Blaine and T. Perkis. The jam-o-drum interactive music system : a study in interaction design. In *DIS '00 : Proceedings of the 3rd conference on Designing interactive systems*, pages 165–173, New York, NY, USA, 2000. ACM.
- [16] B. Bongers. Exploring novel ways of interaction in musical performance. In *C&C '99 : Proceedings of the 3rd conference on Creativity & cognition*, pages 76–81, New York, NY, USA, 1999. ACM.
- [17] B. Bongers. Physical interfaces in the electronic arts. *Trends in Gestural Control of Music*, pages 41–70, 2000.
- [18] J. N. Bott, J. G. Crowley, and J. J. LaViola, Jr. Exploring 3d gestural interfaces for music creation in video games. In *FDG '09 : Proceedings of the 4th International Conference on Foundations of Digital Games*, pages 18–25, New York, NY, USA, 2009. ACM.
- [19] L. Bouguila, M. Ishii, and M. Sato. Scaleable spidar : A haptic interface for human-scale virtual environments. In *Proceedings of the First International Workshop on Haptic Human-Computer Interaction*, pages 182–193, London, UK, 2001. Springer-Verlag.
- [20] D. Bowman, D. Koller, and L. F. Hodges. Travel in immersive virtual environments : An evaluation of viewpoint motion control techniques. In *IEEE Proceedings of VRAIS'97*, 1997.
- [21] D. A. Bowman. Interaction techniques for immersive virtual environments : Design, evaluation, and application. *Journal of Visual Languages and Computing*, 10 :37–53, 1998.
- [22] 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.
- [23] D. A. Bowman, E. Kruijff, J. J. LaViola, and I. Poupyrev. *3D User Interfaces : Theory and Practice*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 2004.
- [24] D. A. Bowman and C. A. Wingrave. Design and evaluation of menu systems for immersive virtual environments. *Virtual Reality Conference, IEEE*, 0 :149, 2001.

- [25] D. A. Bowman, C. A. Wingrave, J. M. Campbell, and V. Q. Ly. Using pinch gloves for both natural and abstract interaction. In *Proceedings of HCI International*, pages 629–633, 2001.
- [26] A. Bregman. *Auditory Scene Analysis : The Perceptual Organization of Sound*. Bradford Books, MIT Press., 1990.
- [27] F. P. Brooks, Jr., M. Ouh-Young, J. J. Batter, and P. Jerome Kilpatrick. Project gropehaptic displays for scientific visualization. In *SIGGRAPH '90 : Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pages 177–185, New York, NY, USA, 1990. ACM.
- [28] C. Cadoz. *Musique, geste, technologie*. Éditions Parenthèses, 1999.
- [29] C. Cadoz, A. Luciani, and J. Florens. CORDIS-ANIMA : a Modeling and simulation system for sound and image synthesis : the general formalism. *Computer music journal*, 17(1) :19–29, 1993.
- [30] J. L. Caivano. Colour and sound : Physical and psychophysical relations. *Color Research and Application*, 19(2), pages 126–132, 1994.
- [31] K. Cascone. Laptop music - counterfeiting aura in the age of infinite reproduction. *Parachute*, issue 107, 2002.
- [32] A. Chaudhary and A. Freed. Visualization, Editing and Spatialization of Sound Representations using the OSE Framework. *Audio engineering society*, 1999.
- [33] M. Chion. *Audio-Vision : Sound on Screen*. Columbia University Press, 1994.
- [34] J. M. Chowning. The synthesis of complex audio spectra by means of frequency modulation. pages 526–534, 1973.
- [35] G. Cirio, M. Marchal, T. Regia-Corte, and A. Lécuyer. The magic barrier tape : a novel metaphor for infinite navigation in virtual worlds with a restricted walking workspace. In *VRST '09 : Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology*, pages 155–162, New York, NY, USA, 2009. ACM.
- [36] A. Cockburn. Revisiting 2d vs 3d implications on spatial memory. In *AUIC '04 : Proceedings of the fifth conference on Australasian user interface*, pages 25–31, Darlinghurst, Australia, Australia, 2004. Australian Computer Society, Inc.
- [37] A. Cockburn and B. McKenzie. Evaluating the effectiveness of spatial memory in 2d and 3d physical and virtual environments. In *CHI '02 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 203–210, New York, NY, USA, 2002. ACM.
- [38] J.-M. Couturier. *Utilisation avancée d'interfaces graphiques dans le contrôle gestuel de processus sonores*. PhD thesis, Université Aix-Marseille II, 2004.
- [39] J.-M. Couturier and M. Roy. Grapholine, instrument audiovisuel de "dessin musical". In *Proceedings of the Journées d'informatique musicale*, 2008.
- [40] C. Cruz-Neira, D. J. Sandin, and T. A. DeFanti. Surround-screen projection-based virtual reality : the design and implementation of the cave. In *SIGGRAPH '93 : Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 135–142, New York, NY, USA, 1993. ACM.

- [41] 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.
- [42] 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, France, 2006. IRCAM — Centre Pompidou.
- [43] M. Desainte Catherine and S. Marchand. Structured Additive Synthesis : Towards a Model of Sound Timbre and Electroacoustic Music Forms. In *Proceedings of the International Computer Music Conference (ICMC99)*, pages 260–263, Chine, 10 1999.
- [44] C. Dobrian and D. Koppelman. The 'E' in NIME : musical expression with new computer interfaces. In *Proceedings of the 2006 conference on New interfaces for musical expression*, page 282. IRCAM—Centre Pompidou, 2006.
- [45] D. S. Ebert, R. M. Rohrer, C. D. Shaw, P. Panda, J. M. Kukla, and D. A. Roberts. Procedural shape generation for multi-dimensional data visualization. In *Proceedings of Data Visualization '99*, 1999.
- [46] S. Ferguson, A. V. Moereand, and D. Cabrera. Seeing sound : Real-time sound visualisation in visual feedback loops used for training musicians. In *Ninth International Conference on Information Visualisation (IV'05)*, 2005.
- [47] G. W. Fitzmaurice, H. Ishii, and W. A. S. Buxton. Bricks : laying the foundations for graspable user interfaces. In *CHI '95 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 442–449, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.
- [48] J. Florens, A. Luciani, C. Cadoz, and N. Castagné. ERGOS : A multi-degrees of freedom and versatile force-feedback panoply. In *Proceedings of EuroHaptics*, pages 356–360, 2004.
- [49] E. Franco, N. J. L. Griffith, and M. Fernström. Issues for designing a flexible expressive audiovisual system for real-time performance & composition. In *NIME '04 : Proceedings of the 2004 conference on New interfaces for musical expression*, pages 165–168, Singapore, Singapore, 2004. National University of Singapore.
- [50] B. Fröhlich, R. Blach, O. Stefani, J. Hochstrate, J. Hoffmann, K. Klüger, and M. Bues. Implementing multi-viewer stereo displays. In *Proceedings of WSCG*. Citeseer, 2005.
- [51] P. Fuchs. *Les interfaces de la réalité virtuelle*. Les Presses de l'Ecole des Mines de Paris, 1996.
- [52] D. Gerber and D. Bechmann. Dogmerv : A geometric deformation modeller for the workbench. *6th Virtual Reality international conference IEEE-VRIC 2004*, pages 165–172, 2004.
- [53] D. Gerber and D. Bechmann. The spin menu : A menu system for virtual environments. *Virtual Reality Conference, IEEE*, 0 :271–272, 2005.

- [54] K. Giannakis. A comparative evaluation of auditory-visual mappings for sound visualisation. *Organised Sound*, 11(3) :297–307, 2006.
- [55] K. Giannakis and M. Smith. Towards a theoretical framework for sound synthesis based on auditory-visual associations. pages 87–92, 2000.
- [56] K. Giannakis and M. Smith. Imaging soundscapes : Identifying cognitive associations between auditory and visual dimensions. *Musical Imagery*, pages 161–179, 2001.
- [57] J. J. Gibson. *The Theory of Affordances*. Lawrence Erlbaum, 1977.
- [58] D. M. Green. Profile analysis : Auditory intensity discrimination. *Oxford psychology series, No. 13*, page 138, 1988.
- [59] M. Hachet, F. Declé, S. Knodel, and P. Guitton. Navidget for easy 3d camera positioning from 2d inputs. In *3DUI '08 : Proceedings of the 2008 IEEE Symposium on 3D User Interfaces*, pages 83–89, Washington, DC, USA, 2008. IEEE Computer Society.
- [60] M. Hachet, P. Guitton, and P. Reuter. The cat for efficient 2d and 3d interaction as an alternative to mouse adaptations. In *VRST '03 : Proceedings of the ACM symposium on Virtual reality software and technology*, pages 225–112, New York, NY, USA, 2003. ACM.
- [61] R. Hamilton. Maps and legends : Designing fps-based interfaces for multi-user composition, improvisation and immersive performance. *Computer Music Modeling and Retrieval. Sense of Sounds : 4th International Symposium, CMMR 2007, Copenhagen, Denmark, August 27-31, 2007. Revised Papers*, 2008.
- [62] R. Hamilton. q3osc : or how i learned to stop worrying and love the game. In *Proceedings of the International Computer Music Association Conference*, 2008.
- [63] C. G. Healey. Building a perceptual visualisation architecture, 2000.
- [64] C. G. Healey. Formalizing artistic techniques and scientific visualization for painted renditions of complex information spaces. In *Proceedings IJCAI 2001*, 2001.
- [65] C. G. Healey, K. S. Booth, and J. T. Enns. High-speed visual estimation using preattentive processing. *ACM Trans. Comput.-Hum. Interact.*, 3(2) :107–135, 1996.
- [66] R. Hiraga, R. Mizaki, and I. Fujishiro. Performance visualization : a new challenge to music through visualization. In *MULTIMEDIA '02 : Proceedings of the tenth ACM international conference on Multimedia*, pages 239–242, New York, NY, USA, 2002. ACM.
- [67] A. Hunt and R. Kirk. Mapping strategies for musical performance. *Trends in Gestural Control of Music*, pages 231–258, 2000.
- [68] A. Hunt and R. Kirk. Midgrid : past, present and future. In *NIME '03 : Proceedings of the 2003 conference on New interfaces for musical expression*, pages 135–139, Singapore, Singapore, 2003. National University of Singapore.

- [69] C. Jacquemin, R. Ajaj, R. Cahen, Y. Ollivier, and D. Schwarz. Plumage : Design d'une interface 3d pour le parcours d'échantillons sonores granularisés. In *Proceedings of the Conférence Francophone sur l'Interaction Homme-Machine(IHM'07)*, 2007.
- [70] K. Jensen. *Timbre models of musical sounds*. PhD thesis, Department of Computer Science, University of Copenhagen, 1999.
- [71] S. Jordà. Fmol : Toward user-friendly, sophisticated new musical instruments. *Computer Music Journal*, 26(3) :23–39, 2002.
- [72] S. Jordà. *Crafting musical computers for new musics' performance and improvisation*. PhD thesis, Universitat Pompeu Fabra, 2005.
- [73] S. Jordà, M. Kaltenbrunner, G. Geiger, and R. Bencina. The reactable\*. In *Proceedings of the International Computer Music Conference*, 2005.
- [74] Y. Kitamura, T. Konishi, S. Yamamoto, and F. Kishino. Interactive stereoscopic display for three or more users. In *SIGGRAPH '01 : Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 231–240, New York, NY, USA, 2001. ACM.
- [75] J. Krimphoff, S. McAdams, and S. Winsberg. Caractérisation du timbre des sons complexes. ii : Analyses acoustiques et quantification psychophysique. *Journal de Physique 4*, pages (C5) :625–628, 1994.
- [76] M. W. Krueger, T. Gionfriddo, and K. Hinrichsen. Videoplace—an artificial reality. *SIGCHI Bull.*, 16(4) :35–40, 1985.
- [77] M. Kubovy and D. V. Valkenburg. Auditory and visual objects. *Cognition*, 80(1-2) :97 – 126, 2001. Objects and Attention.
- [78] A. Lécuyer, J.-M. Burkhardt, and L. Etienne. Feeling bumps and holes without a haptic interface : the perception of pseudo-haptic textures. In *CHI '04 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 239–246, New York, NY, USA, 2004. ACM.
- [79] E. Lemi and A. Georgaki. Reviewing the transformation of sound to image in new computer music software. In *Proceedings of the 4th Sound and Music Computing Conference*, 2007.
- [80] G. Levin. *Painterly Interfaces for Audiovisual Performance*. PhD thesis, 2000.
- [81] S. D. Lipscomb and E. M. Kim. Perceived match between visual parameters and auditory correlates : an experimental multimedia investigation. In *Proceedings of the 8th International Conference on Music Perception and Cognition*, 2004.
- [82] S. J. Luck and E. K. Vogel. The capacity of visual working memory for features and conjunctions. *Nature*, Vol 390, 1997.
- [83] T. Magnusson. Screen-based musical interfaces as semiotic machines. In *Proceedings of the 2006 International Conference on New Interfaces for Musical Expression (NIME06)*, Paris, France, 2006.
- [84] 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.

- [85] R. Marczak. Etude d'une représentation hiérarchique liant micro et macro-structures musicales. Master's thesis, University of Bordeaux, 2007.
- [86] M. T. Marshall and M. M. Wanderley. Vibrotactile feedback in digital musical instruments. In *NIME '06 : Proceedings of the 2006 conference on New interfaces for musical expression*, pages 226–229, Paris, France, France, 2006. IRCAM — Centre Pompidou.
- [87] T. M. MASSIE. The phantom haptic interface : A device for probing virtual objects. *Proceedings of ASME Dynamic Systems and Control Division*, 55 :295–301, 1994.
- [88] C. McCormick. Ergates : <http://mccormick.cx/projects/ergates/>.
- [89] P. McLeod, J. Driver, and J. Crisp. Visual search for a conjunction of movement and form is parallel. *Nature*, 332 :154–155, 1988.
- [90] D. K. Mellinger. *Event Formation and Separation in Musical Sound*. PhD thesis, Stanford University, CA, USA, 1991.
- [91] N. Moody. Motion as the connection between audio and visuals, 2006.
- [92] R. A. Moog. Midi : Musical instrument digital interface. *J. Audio Eng. Soc.*, 34(5) :394–404, 1986.
- [93] A. M. Treisman. A feature-integration theory of attention. *Cognitive Psychology*, vol 12, pages 97–136, 1980.
- [94] A. Mulder. Virtual musical instruments : Accessing the sound synthesis universe as a performer. In *Proceedings of the First Brazilian Symposium on Computer Music*, pages 243–250, 1994.
- [95] A. G. Mulder. *Design of virtual three-dimensional instruments for sound control*. PhD thesis, Simon Fraser University, Canada, 1998.
- [96] M. Naef and D. Collicot. A vr interface for collaborative 3d audio performance. In *Proceedings of the 2006 International Conference on New Interfaces for Musical Expression (NIME06)*, Paris, France, 2006.
- [97] J. Olive and S. Pickles. q3apd : <http://selectparks.net/julian/q3apd/>.
- [98] J. Oliver and S. Pickles. Fijuu2 : a game-based audio-visual performance and composition engine. In *NIME '07 : Proceedings of the 7th international conference on New interfaces for musical expression*, pages 430–430, New York, NY, USA, 2007. ACM.
- [99] M. S. O'Modhrain. *Playing by feel : incorporating haptic feedback into computer-based musical instruments*. PhD thesis, Stanford, CA, USA, 2001. Adviser-Chafe, Chris.
- [100] I. Poupyrev, S. Weghorst, M. Billinghurst, and T. Ichikawa. *Egocentric object manipulation in virtual environments : Empirical evaluation of interaction techniques*. 1998.
- [101] J. Pressing. Some perspectives on performed sound and music in virtual environments. *Presence*, 6(4) :482–503, 1997.

- [102] M. S. Puckette, T. Apel, and D. D. Zicarelli. Real-time audio analysis tools for pd and msp. In *Proceedings of the International Computer Music Conference*, 1998.
- [103] V. Pulkki. Virtual sound source positioning using vector base amplitude panning. pages 456–466, 1997.
- [104] T. Rahne, M. Böckmann, H. von Specht, and E. S. Sussman. Visual cues can modulate integration and segregation of objects in auditory scene analysis. *Brain Research*, 1144 :127 – 135, 2007.
- [105] J. Reuter. Soundpaint – painting music. In *Proceedings of the 3rd International Linux Audio Conference*, 2005.
- [106] C. Roads. Automated granular synthesis of sound. pages 61–62, 1978.
- [107] 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.
- [108] A. Sedes, B. Courribet, and J.-B. Thiebaut. Visualization of sound as a control interface. In *Proceedings of the 7th International Conference on Digital Audio Effects (DAFX-04)*, 2004.
- [109] B. Shneiderman. Direct manipulation : A step beyond programming languages (abstract only). In *CHI '81 : Proceedings of the joint conference on Easier and more productive use of computer systems. (Part - II)*, page 143, New York, NY, USA, 1981. ACM.
- [110] M. Slater. Measuring presence : A response to the witmer and singer presence questionnaire. *Presence : Teleoper. Virtual Environ.*, 8(5) :560–565, 1999.
- [111] J. Smith III. Physical modeling using digital waveguides. *Computer Music Journal*, 16(4) :74–91, 1992.
- [112] S. S. Stevens. A scale for the measurement of a psychological magnitude : loudness. *Psychological Review. Vol 43(5)*, pages 405–416, 1936.
- [113] I. Sutherland. *Sketchpad-a Graphical Man-Machine Interface*. PhD thesis, 1963.
- [114] R. M. Taylor, II, T. C. Hudson, A. Seeger, H. Weber, J. Juliano, and A. T. Helsen. 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.
- [115] 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, Singapore, 2005. National University of Singapore.
- [116] M. Usoh, K. Arthur, M. C. Whitton, R. Bastos, A. Steed, M. Slater, and F. P. Brooks, Jr. Walking , walking-in-place , flying, in virtual environments. In *SIGGRAPH '99 : Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 359–364, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

- [117] L. Valbom and A. Marcos. Wave : Sound and music in an immersive environment. *Computer and Graphics*, 2005.
- [118] L. Valbom and A. Marcos. An immersive musical instrument prototype. *IEEE Comput. Graph. Appl.*, 27(4) :14–19, 2007.
- [119] V. Verfaillie. *Effets audio-numériques adaptatifs*. PhD thesis, 2003.
- [120] B. Verplank, M. Mathews, and R. Shaw. Scanned synthesis. *The Journal of the Acoustical Society of America*, 109(5) :2400–2400, 2001.
- [121] R. Vertegaal, T. Ungvary, and M. Kieslinger. Towards a musician’s cockpit : transducers, feedback and musical function. In *Proceedings of the International Computer Music Conference*, pages 308–311. Citeseer, 1996.
- [122] H. Vinet. *Interfaces homme-machine et création musicale*, chapter Concepts d’interfaces graphiques pour la production musicale et sonore. Hermès Sciences, 1999.
- [123] J. Volkman, S. S. Stevens, and E. B. Newman. A scale for the measurement of the psychological magnitude pitch. *The Journal of the Acoustical Society of America*, 8(3) :208–208, 1937.
- [124] G. von Békésy, E. G. Wever, and W. T. Peake. Experiments in hearing. *The Journal of the Acoustical Society of America*, 88(6) :2905–2905, 1990.
- [125] M. Waisvisz. The hands, a set of remote midi-controllers. In *Proceedings of the International Computer Music Conference(ICMC’85)*, pages 313–318, 1985.
- [126] R. Walker. The effects of culture, environment, age, and musical training on choices of visual metaphors for sound. *Perception and Psychophysics Vol 42(5)*, pages 491–502, 1987.
- [127] M. Wanderley, N. Orió, and N. Schnell. Towards an analysis of interaction in sound generating systems. In *ISEA2000 Conference Proceedings*, 2000.
- [128] M. M. Wanderley and P. Depalle. Gestural control of sound synthesis. In *Proceedings of the IEEE*, pages 632–644, 2004.
- [129] C. Ware. Color sequences for univariate maps : Theory, experiments and principles. *IEEE Computer Graphics and Applications*, 8 :41–49, 1988.
- [130] C. Ware. *Information Visualization : Perception for Design*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.
- [131] C. Ware and W. Knight. Using visual texture for information display. In *ACM Transactions on Graphics*, 14(1) :3–20, 1995.
- [132] G. Weinberg, R. Aimi, and K. Jennings. The beatbug network : a rhythmic system for interdependent group collaboration. In *NIME ’02 : Proceedings of the 2002 conference on New interfaces for musical expression*, pages 1–6, Singapore, Singapore, 2002. National University of Singapore.
- [133] D. Wessel and M. Wright. Problems and prospects for intimate musical control of computers. *Computer Music Journal*, 26(3) :11–22, 2002.
- [134] J. Whitney. *Digital Harmony : on the Complementarity of Music and Visual Art*. McGraw-Hill Inc., 1980.

- [135] M. Wozniowski, Z. Settel, and J. Cooperstock. A spatial interface for audio and music production. In *Proceedings of the International Conference on Digital Audio Effects (DAFx), 2006*, 2006.
- [136] M. Wright. Open sound control : an enabling technology for musical networking. *Org. Sound*, 10(3) :193–200, 2005.
- [137] S. Zhai, P. Milgram, and W. Buxton. The influence of muscle groups on performance of multiple degree-of-freedom input. In *CHI '96 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 308–315, New York, NY, USA, 1996. ACM.
- [138] H. Zhang and E. D. Montag. Perceptual color scales for univariate and bivariate data display. In *Proceedings of the 30th International Congress of Imaging Science ICIS06*, 2006.
- [139] E. Zwicker. Subdivision of the audible frequency range into critical bands. *The Journal of the Acoustical Society of America*, vol. 33, page 248, 1961.

# Appendices



# Syntaxe DRILE

```
<!--Les fichiers *.dru definissent la configuration de l application
      Drile-ui, qui gere le rendu graphique de Drile-->
<drile-ui>
  <!-- Port d entree OSC et Entrees VRPN -->
  <input osc_input_port="7000" vrpn_input_tracker="DTrack@hel.labri.fr"
    />

  <!--Definition des destinations des messages OSC-->
  <osc_output_socket name="socket1" address="127.0.0.1" port="7001"/>

  <!--Definition d une camera avec notamment le mode de rendu
      stereoscopique, la position de l ecran dans la scene-->
  <camera name="cam1" stereo="none" connection_type="StreamSock"
    connection_address="127.0.0.1" connection_port="7010"
    eye_separation="0.06" number_of_displays="1" x_overlap="0"
    bottom_left="-2;0;-3" bottom_right="2;0;-3" top_right="2;3;-3"
    top_left="-2;3;-3" head_bounds=" -1.5/1.5/1.2/1.6/ -3/0.2">

    <!--Les balises parameter representent des parametres qui recoivent
      et/ou envoient des messages VRPN ou OSC-->

    <!--Les parametres hpos et hrot permettent de controler la position
      de la tete de l utilisateur pour le suivi de tete-->
    <parameter name="hposX" value="0">
      <input type="vrpn" vrpn_type="tracker" vrpn_device="1" vrpn_param
        ="0"/>
      <input type="osc"/>
    </parameter>
    <parameter name="hposY" value="0"/>
    <parameter name="hposZ" value="0"/>
    <parameter name="hrotX" value="0"/>
    <parameter name="hrotY" value="0"/>
    <parameter name="hrotZ" value="0"/>

    <!--Les parametres pos et rot permettent de controler la position
      de la camera-->
    <parameter name="posX" value="0"/>
    <parameter name="posY" value="0"/>
    <parameter name="posZ" value="0"/>
    <parameter name="rotX" value="0"/>
    <parameter name="rotY" value="0"/>
    <parameter name="rotZ" value="0"/>
  </camera>
```

```

<!--Definition d une lumiere , elle peut etre attachee a une camera ou
fixe-->
<light name="light1" attached_to="nothing">
  <parameter name="posX" value="0">
    <input type="osc"/>
  </parameter>
</light>

<!--Definition d une scene-->
<scene name="scene2" coords="0/1/0" osc_output_socket="socket1">

  <!--Definition des worms avec leur noms et leur positions-->
  <worm name="worm5" pos_x="-1.2" pos_y="1"/>
  <worm name="worm6" pos_x="-0.4" pos_y="1"/>
  <worm name="worm7" pos_x="0.4" pos_y="1" />
  <worm name="worm8" pos_x="1.2" pos_y="1" />

  <!--Definition des tunnels avec leurs positions , parametres et
  echelles-->
  <tunnel pos_x="-1" pos_y="2">
    <control parameter="scaleY">
      <preset type="array" values="0;0.15;0.45;0.75;1"/>
      <preset type="linear" values="0;1"/>
    </control>
  </tunnel>
  <tunnel pos_x="0.5" pos_y="1.5">
    <control parameter="colorHue">
      <preset type="linear" values="0.1;1"/>
    </control>
  </tunnel>

  <!--Definition de l apparence de la piece , en utilisant des
  motifs de formes et de variations de parametres-->
  <room>
    <preset shapes_pattern="B">
      <parameter name="colorHue" range1="0.2;1" range2="0.2;0"
      pattern="1;3" input="high"/>
    </preset>
  </room>
</scene>

<!--Definition de Piivert , main droite puis main gauche , ici la
position et l orientation sont controlables par VRPN et OSC-->
<tool name="piivert1R" type="piivert" attached_to="cam1">
  <parameter name="grab" value="0">
    <input type="osc"/>
  </parameter>
  <parameter name="loop" value="0">
    <input type="osc"/>
  </parameter>
  <parameter name="posX" value="0">
    <input type="osc"/>
    <input type="vrpn" vrpn_type="tracker" vrpn_device="4"
    vrpn_param="0"/>
  </parameter>

```

```

<!--Ici la rotation est definie par un quaternion-->
<parameter name="rotQ1" value="0">
  <input type="osc"/>
  <input type="vrpn" vrpn_type="tracker" vrpn_device="4"
    vrpn_param="6"/>
</parameter>
<parameter name="rotQ2" value="0"/>
<parameter name="rotQ3" value="0"/>
<parameter name="rotQ4" value="0"/>

<!--Ces parametres sont utilises pour la communication avec drile
-audio-->
<parameter name="ray1selected">
  <output type="osc" osc_socket="socket1"/>
</parameter>
<parameter name="ray1hit">
  <input type="osc"/>
</parameter>
<parameter name="ray1scratch">
  <input type="osc"/>
</parameter>
</tool>

  <tool name="piivert1L" type="piivert" attached_to="cam1">
    <parameter ...
    ...
  </parameter>
</tool>
</drile-ui>

<!--Les fichiers *.dra definissent la configuration de l application
Drile-audio, qui gere le rendu sonore de Drile-->
<drile-audio>
  <!--Definition des connexions OSC pour la communication avec drile-
audio-->
    <connections input_port="7001" output_adress="127.0.0.1" output
      \_port="7000"/>

  <!--Definition des numeros des controles MIDI envoyes par les
capteurs-->
    <piivert name="piivert1" right_hand_sensors="
      40/32;41/33;42/34;43/35;" left_hand_sensors="
      44/36;45/37;46/38;47/39;" >

  <!--Suite d effets et mappings avec les parametres graphiques-->
    <effect plugin="tempo" graphic="rotationSpeed"/>
    <effect plugin="reverb" graphic="scattering"/>
    <effect plugin="pitch" graphic="colorHue"/>
    <effect plugin="distortion" graphic="transparency"/>
    <effect plugin="volume" graphic="scaleY"/>

  <!--Analyses et parametres graphiques associes-->
    <analysis plugin="spectrum" graphic="shape"/>
    <analysis plugin="readpointer" graphic="rotation"/>

```

```
<!--Definition d une scene-->
  <scene name="scene1">

  <!--Worms avec leurs modes et leur fichiers audio-->
    <worm name="worm1" polyphony="3" mode="trigger">
      <sample file="sons-drile/cc134.wav"/>
    </worm>
    <worm name="worm2" polyphony="3" mode="trigger">
      <sample file="sons-drile/gc34.wav"/>
    </worm>
    <worm name="worm3" polyphony="3" mode="trigger">
      <sample file="sons-drile/charle3.wav"/>
    </worm>
    <worm name="worm4" polyphony="3" mode="trigger">
      <sample file="sons-drile/md3.wav"/>
    </worm>
  </scene>
</drile-audio>
```