

Mémoire de Master recherche  
Année 2010-2011

ENCADRANTS :  
Julien Castet  
Myriam Desainte-Catherine  
Martin Hachet

# **Construction opportuniste de la musique**

**Jonathan Aceituno**

# Table des matières

Chapitre 1	<b>Introduction</b> . . . . .	<b>3</b>
	1. Sujet et motivations . . . . .	3
	2. Organisation du document . . . . .	3
Chapitre 2	<b>Contexte</b> . . . . .	<b>4</b>
	1. Informatique tangible : l'informatique du monde réel . . . . .	4
	<i>Interaction homme-machine • Du virtuel au réel • Interaction tangible • Mise en commun des styles d'interaction prenant en compte le réel • Vers une vision située de l'interaction • Conclusion.</i>	
	2. Interaction tangible pour la musique . . . . .	9
	<i>Interaction musicale • Environnements graphiques pour la musique • Interaction tangible et musique • Conclusion.</i>	
Chapitre 3	<b>Problématique</b> . . . . .	<b>12</b>
	1. Construction musicale dans les interfaces tangibles . . . . .	12
	<i>Construction et performance musicale • Utilisations de la reconfiguration dynamique • Construction musicale tangible.</i>	
	2. Opportunisme . . . . .	16
	<i>Définition • Positionnement • Caractérisation • Vers une construction musicale opportuniste . . . . .</i>	
	3. Formulation de la problématique . . . . .	19
Chapitre 4	<b>Démarche</b> . . . . .	<b>21</b>
	1. Choix des capteurs pour l'interaction opportuniste . . . . .	21
	<i>Contraintes • Importance de la vision • Multimodalité de la captation • Conclusion.</i>	
	2. Segmentation pour le suivi par marqueurs . . . . .	24
	<i>Méthode • Suivi par marqueurs • Segmentation • Association et exploitation • Conclusion.</i>	
	3. Couplage virtuel . . . . .	29
	<i>Motivations • Modèle physique • Méthode • Conclusion.</i>	
	4. Techniques d'interaction opportunistes . . . . .	31
	<i>Motivations • Techniques pour la liaison dynamique opportuniste • Techniques pour la composition dynamique opportuniste • Méthode • Conclusion.</i>	
Chapitre 5	<b>Résultats</b> . . . . .	<b>36</b>
	1. Un système de construction opportuniste . . . . .	36
	<i>Environnement, processus musicaux et comportements • Gestion transparente des processus musicaux • Conception des comportements.</i>	
	2. Implémentation . . . . .	39
	<i>Programme de synthèse sonore • Programme principal . . . . .</i>	
	3. Scénarios et observations . . . . .	44
	<i>Description des processus sonores et comportements implémentés . . . . . • Scénarios.</i>	
	4. Perspectives . . . . .	48
	<i>Limites du modèle actuel • Prototypage et évaluation des instruments numérique.</i>	
Chapitre 6	<b>Conclusion</b> . . . . .	<b>51</b>

# Introduction

## 1 Sujet et motivations

Ce stage s'inscrit dans la continuité des travaux menés sur le contrôle opportuniste de la musique[54]. L'originalité de ces travaux est de considérer l'environnement immédiat comme un outil de contrôle pour la performance musicale : une règle graduée peut servir pour régler le volume d'une séquence de batterie, taper sur un rouleau de bande adhésive peut lancer cette séquence, etc. Cette idée offre d'énormes possibilités d'interaction originales, mais la séparation des modalités de construction et des modalités de contrôle des structures qui produisent le son limite de façon significative la maniabilité des systèmes mis en œuvre. Réunir la construction et le contrôle des processus sonores dans un cadre opportuniste paraît être une voie prometteuse pour améliorer certains aspects de l'interaction musicale. Dans ce stage, nous adoptons une démarche exploratoire, nous intéressons particulièrement à la problématique de la construction, et proposons quelques pistes pour évoluer dans cette voie, dans le cadre plus général du contrôle musical.

Nos objectifs pour ce stage sont de :

- présenter les notions qui permettent de comprendre les problématiques et les enjeux de l'interaction musicale, et plus particulièrement de l'interaction tangible pour la musique ;
- définir les notions de construction musicale et d'opportunisme, et de voir comment elles s'expriment à travers l'interaction tangible pour la musique dans le concept de construction musicale opportuniste ;
- analyser les problèmes techniques qui surviennent dans le cadre de la construction musicale opportuniste et proposer des solutions ;
- et finalement, proposer un modèle de construction musicale opportuniste illustré à travers un système interactif qui présente les solutions proposées.

## 2 Organisation du document

Ce document est organisé de la manière suivante. Le chapitre 2 plante tout d'abord le contexte de l'interaction homme-machine et de l'interaction musicale. Le chapitre 3 introduit les notions de construction musicale et d'opportunisme, propose d'unir ces deux notions dans le terme de construction musicale opportuniste, ce qui définit la problématique centrale du document, et présente les implications techniques de cette union. Le chapitre 4 propose des solutions aux problèmes techniques évoqués. Enfin, le chapitre 5 propose un modèle conceptuel de construction musicale opportuniste et décrit la réalisation d'un prototype réalisant ce modèle et illustrant le travail réalisé.

## 1 Informatique tangible : l'informatique du monde réel

Les ordinateurs incarnent le traitement automatique, qui rend possible la résolution d'une grande variété de problèmes. Depuis le début de l'informatique, il a toujours été difficile de faire usage de ce pouvoir magique, et bien que l'interaction entre les humains et les machines se soit améliorée, elle évoque toujours un sentiment d'*amour-baine* chez la plupart des utilisateurs. Après une revue des enjeux de l'interaction homme-machine, nous constaterons une volonté d'extirper le traitement automatique de l'ordinateur pour l'intégrer dans le monde qui nous entoure. Nous décrirons quelles sont les contributions de l'interaction tangible et enfin comment ce style d'interaction et d'autres sont étudiés pour répondre à cette vision de l'informatique qui fusionne avec le monde qui nous entoure.

### 1. Interaction homme-machine

Initialement considérée comme la science du traitement automatique des informations, l'informatique a dû voir au-delà du calcul pur et simple et prendre en compte des situations d'interaction, voire de communication, avec le monde extérieur. Les systèmes fermés qui traitaient des informations par lots se sont transformés en des systèmes ouverts qui changent leur comportement en fonction de la situation évaluée. Un *système interactif* est un système informatique qui extrait et renvoie des informations à un environnement extérieur qu'il ne contrôle pas[82].

Les problèmes que soulève l'interaction avec les systèmes informatiques sont étudiés dans le cadre du domaine de l'*interaction homme-machine* depuis les années 40[38]. Les premières recherches datent de la première génération des ordinateurs munis de tubes à vide, comme l'ENIAC, et visaient à améliorer l'efficacité du travail des opérateurs qui devaient charger les programmes écrits dans l'ordinateur en poussant des boutons et en branchant des câbles.

L'arrivée de la programmation constitua plus tard une première étape importante de l'évolution des interactions avec les systèmes informatiques, en apportant, avec les langages et les compilateurs, la possibilité d'employer des abstractions de haut niveau pour communiquer ses intentions à la machine. Plus tard, les télétypes permettaient enfin un semblant de dialogue, portant l'interactivité des systèmes à un niveau plus proche du temps réel, poussant toujours plus loin l'efficacité des utilisateurs de ces systèmes, réservés alors à l'industrie.

L'informatique graphique et la puissance des nouveaux systèmes a conduit, vers les années 70, à chercher à rendre accessibles les systèmes informatiques. Les interfaces graphiques et l'omniprésence du style WIMP<sup>1</sup> aujourd'hui sont dues à des travaux comme ceux sur l'ordinateur personnel Star au PARC de Xerox. Ces interfaces dépassent l'aspect conversationnel de l'interaction textuelle en proposant d'utiliser l'espace visuel pour interagir avec

---

<sup>1</sup> *Windows, icons, menus and pointing devices*, qui composent l'essentiel des modalités d'interaction que nous, à l'exception des vrais barbus, employons avec les ordinateurs.

l'information, s'appuyant sur les compétences des personnes pour l'organisation et le repérage spatial.

Aujourd'hui, l'interaction homme-machine étudie l'interaction entre les systèmes interactifs et les humains, ces fameux *utilisateurs*, en cherchant comment concevoir, réaliser et évaluer l'interaction à travers ces systèmes interactifs. C'est l'amélioration de l'interaction qui est visée ici plutôt que seulement celle des *interfaces*. Ces dernières ne sont que les moyens mis en œuvre pour supporter l'interaction. Sur un air de Dijkstra, Beaudouin-Lafon nous précise que l'interaction homme-machine n'est pas plus la science des interfaces que l'astronomie est celle des télescopes[5].

## 2. Du virtuel au réel

De nos jours<sup>2</sup>, les interfaces graphiques sont les principaux moyens d'utiliser un ordinateur. L'interaction graphique se fait en comprenant ses paradigmes, ses codes, ses protocoles : les gens qui n'ont jamais eu affaire à une de ces machines ne savent pas comment les utiliser. Certains de ces codes, comme la manipulation directe, sont inspirés de notre réalité, tandis que d'autres sont totalement nouveaux et demandent un temps d'apprentissage. En plus du monde dans lequel nous vivons, où l'humain peine à acquérir quelques compétences comme la marche ou la parole, il doit maîtriser les compétences et les conventions nécessaires à l'exploitation d'un monde artificiel et intangible.

Par la suite, on qualifiera de virtuels un processus, une structure ou une information qui n'existe qu'au sein d'un système informatique. Le virtuel représente ce que l'ordinateur manipule et qui est intangible pour les humains, il s'oppose au réel qui caractérisera ce qui fait partie du monde que nous percevons, ce qui est tangible — nous nous contenterons de ces définitions et ne les détaillerons pas plus. On peut voir les systèmes interactifs comme des passerelles entre le réel et le virtuel. Un tel système interprète les faits réels qu'il perçoit pour en tirer une représentation virtuelle, pouvant influencer sur d'autres éléments virtuels, dont certains sont traduits en faits réels et perceptibles, comme l'affichage sur un écran. Dans cette vision, on peut alors interpréter les difficultés de concevoir une interaction compréhensible et intuitive dans les interfaces graphiques comme le résultat du décalage entre les notions du réel et celles du virtuel.

Rekimoto et Nagao pointent ce décalage dans des styles d'interaction comme les interfaces graphiques traditionnelles[76] : il y a un fossé entre l'interaction de l'utilisateur avec le monde réel et l'interaction de l'utilisateur avec le monde virtuel. Ils expliquent comment certains autres styles d'interaction comblent ce fossé. La réalité virtuelle cherche à obtenir l'immersion la plus totale de l'utilisateur dans le monde virtuel en coupant toute communication avec le monde réel. La réalité augmentée propose une approche différente, en permettant à l'ordinateur d'être le médiateur de certaines interactions entre le monde réel et l'utilisateur, augmentant par exemple les objets environnants en leur rajoutant visuellement des informations correspondant aux souhaits qu'a émis l'utilisateur à travers l'interface. L'informatique ubiquitaire voit les systèmes informatiques et leurs interfaces se fondre dans les autres objets du monde réel et pouvant partager quelques modalités d'interaction[84] : on peut le voir comme la généralisation de la réalité augmentée. Le point commun à ces styles d'interaction, et à d'autres également, est donc l'idée d'une interaction plus directe grâce à la prise en compte par les systèmes informatiques du monde réel dans lequel ils s'inscrivent.

<sup>2</sup> Et plus pour très longtemps, nous disent certains depuis 20 ans...

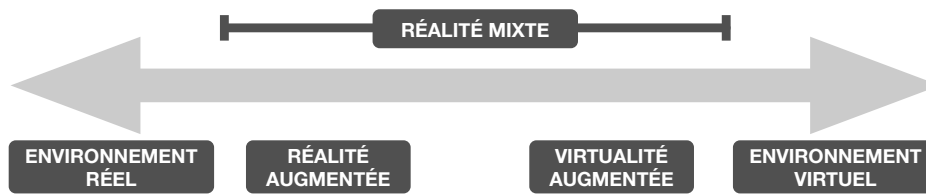
### 3. Interaction tangible

L'idée est poussée encore plus loin lorsqu'il est proposé d'utiliser des objets physiques comme médiateurs de l'interaction. Les interfaces préhensibles, ou *graspable interfaces*[34], sont la première incarnation de ce concept. À la place d'utiliser une souris et un écran, Fitzmaurice préconise qu'il est mieux d'utiliser un ou plusieurs objets physiques spécialisés pour accomplir certaines tâches. Ces interfaces préhensibles présentent plusieurs traits caractéristiques. Chaque objet <sup>3</sup> représente une fonction précise, et chaque moyen d'entrée et de sortie est accessible avec un objet : l'entrée et la sortie sont multiplexés dans l'espace (au contraire, la souris change de fonction au fil du temps, cette entrée est multiplexée dans le temps dans les interfaces graphiques traditionnelles). Les objets peuvent être utilisés de manière concurrente, par exemple, avec les deux mains en même temps (alors que la souris oblige d'ordonner séquentiellement les différentes manipulations). Les fonctions ou les données ont une forme physique préhensible, celle d'un objet qu'on manipulera, qui est spécifique plutôt que générique. Enfin, ces objets réagissent en fonction de leur position dans l'espace par rapport à un référentiel physique qui doit pouvoir être changé rapidement. Les interfaces préhensibles montrent leur utilité dans certaines situations car leurs propriétés font que l'interaction est simplifiée du fait qu'il est possible d'utiliser les compétences de manipulation spatiale des humains et leurs sens haptiques. De plus, elles décroissent la complexité de l'interaction par rapport aux interfaces graphiques en permettant naturellement les actions *épistémiques*, c'est-à-dire les actions qui n'ont aucun sens pour la tâche en cours mais qui servent à l'utilisateur pour se représenter physiquement les problèmes, de façon à réduire sa charge cognitive[65], comme lorsqu'on réorganise ses cartes dans une partie de tarot.

La prochaine étape dans cette voie fut décrite par Ishii et Ullmer. Ces derniers cherchent à rendre l'information virtuelle physique et préhensible (les *tangible bits*) de manière à ce qu'elle fasse partie du monde physique[44], comme objets matériels que nous manipulons lorsqu'ils sont le centre de notre attention (*foreground*), mais aussi éventuellement en agissant à la périphérie de notre perception (*background*) avec la possibilité d'attirer notre attention de manière ambiante, comme la lumière du jour nous renseigne sur le temps qu'il fait. C'est une intégration de plusieurs travaux antérieurs, mais ce furent les interfaces préhensibles qui ont le plus contribué à cette nouvelle vision. *L'interaction tangible* est ainsi née. La notion la plus importante des interfaces tangibles est le couplage entre un objet physique et une information ou une fonctionnalité virtuelle et le fait que cet objet physique est à la fois l'entrée et la sortie pour le système interactif. En complément, Ishii et Ullmer considèrent que les interfaces tangibles peuvent bénéficier de l'utilisation de surfaces, comme une table ou un mur, comme interface entre les mondes réels et virtuels, ainsi que des médias ambiants comme le son ou le vent pour communiquer des informations à la périphérie de notre perception.

Au fur et à mesure que de nombreux systèmes interactifs tangibles étaient réalisés, le besoin de décrire et de caractériser l'interaction tangible se faisait ressentir. MCRpd — modèle, contrôleur, représentation physique et représentation numérique, qui a été renommé plus tard MCRit — est un modèle de l'interaction tangible, inspiré de ce que MVC — modèle, vue et contrôleur — était pour l'interaction graphique traditionnelle. Dans MCRpd, le *modèle* — celui qui fait le M — appartient au monde numérique, tandis que les autres éléments appartiennent au monde physique. L'objet physique exhibe les parties *contrôleur* et

<sup>3</sup> Dans ce mémoire, on utilisera le terme *objet* pour désigner un élément d'une interface tangible. On précisera *objet physique* pour désigner la partie physique de cet élément, *objet virtuel* pour la partie virtuelle, et les autres significations seront indiquées.



**FIGURE 2.1** Continuum de la virtualité, selon [58]

*représentation physique*, tandis que c'est une augmentation de l'objet physique, liée à ce dernier mais n'en faisant pas partie, qui fait la *représentation numérique* [78]. Ce modèle souligne trois caractéristiques fondamentales des interfaces tangibles : le couplage entre l'objet physique et l'information qu'il représente, c'est-à-dire entre la représentation physique et le modèle ; le couplage perçu entre l'objet physique et ses augmentations, c'est-à-dire entre la représentation physique et la représentation numérique ; et enfin le fait que le contrôle est réalisé en manipulant l'objet physique, puisque ce dernier est à la fois le contrôleur et la représentation physique.

De nombreuses autres caractéristiques des interfaces tangibles peuvent être étudiées. On considère par exemple plusieurs types de couplage entre les objets physiques et les informations virtuelles : un objet physique peut représenter un objet virtuel, une personne ou une action. Il serait possible de décrire ces couplage en fonction de deux critères [33] : le degré d'incarnation (*embodiment*), qui détermine à quel point l'utilisateur va considérer que c'est l'objet qui réagit, et la métaphore, qui détermine à quel point l'action de l'utilisateur sur l'objet correspond à l'action qui est transmise à l'objet virtuel correspondant. Enfin, remarquons que certaines interfaces tangibles appuient l'interaction sur des ensembles d'objets selon différentes modalités qui peuvent se mélanger [78] : les systèmes *spatiaux* s'appuient sur la configuration des objets dans l'espace, les systèmes *relationnels* s'appuient sur les relations entre les objets, comme leur ordre ou leur proximité, et les systèmes *constructifs* s'appuient sur l'assemblage des éléments, comme on assemble des blocs LEGO.

#### 4. Mise en commun des styles d'interaction prenant en compte le réel

Il est difficile d'évoquer les propriétés de l'interaction tangible sans évoquer les propriétés d'autres styles d'interaction, car beaucoup de concepts sont communs dans l'informatique ubiquitaire, l'interaction tangible, la réalité virtuelle, la réalité augmentée, la réalité mixte, le *wearable computing*, l'informatique ambiante, etc. La réunion de ces domaines dans ce qu'ils ont de commun a permis de faire émerger plusieurs théories, modèles et taxonomies utiles pour comprendre et concevoir ces systèmes.

À l'issue d'une étude de 44 systèmes se réclamant de l'interaction tangible ou d'un des points du continuum de la virtualité (figure 2.1), Chalon propose une taxonomie riche et le terme unificateur de *réalité mixte*, qui, selon lui, «qualifie un dispositif interactif dans lequel des objets réels et des données informatiques sont mêlés. L'ensemble est perçu de manière cohérente par l'utilisateur par un ou plusieurs de ses sens ce qui lui permet d'interagir avec les différents objets réels ou virtuels de manière cohérente» [15]. Ces propositions lui permettent de développer IRVO, un modèle de conception et de réalisation d'applications de réalité mixte.

Jacob et al. préfèrent considérer que c'est le fait de profiter des compétences des utilisateurs pour évoluer dans l'environnement réel qui réunit les différents styles sous le nom d'*interaction basée sur la réalité* [45], et que ce dont on peut tirer profit se distingue en quatre

**FIGURE 2.1**

thèmes : la connaissance informelle des utilisateurs à propos des principes de la physique (comme la gravité), la compétence qu'ont les utilisateurs avec leur propre corps (comme la préhension manuelle), la connaissance qu'ont les utilisateurs de leur environnement (comme les ombres et les éclairages), et enfin les compétences sociales des utilisateurs (comme la collaboration colocalisée). Ils montrent que les interfaces concernées doivent faire des compromis entre le réalisme apporté par ces quatre thèmes et d'autres caractéristiques comme la puissance expressive, l'efficacité, la versatilité, l'ergonomie, l'accessibilité et l'aspect pratique.

Ces travaux ne font pas de consensus car ils n'abordent pas les mêmes aspects et le choix des termes doit se faire en fonction des besoins et des critères étudiés. Pour ce mémoire, nous utiliserons le terme *informatique tangible* pour évoquer ce qui concerne les systèmes interactifs où les objets sont utilisés comme médiateurs exclusifs de l'interaction.

## 5. Vers une vision située de l'interaction

Les caractéristiques qu'il a été possible de décrire grâce aux travaux évoqués ci-dessus concernent en grande partie l'interaction entre le système et l'utilisateur. Par exemple, Charlon considère que la manipulation des objets dans le cadre de l'interaction tangible a pour seul but d'agir avec les parties virtuelles du système. Or, l'étude et la conception des systèmes interactifs doivent prendre en compte la situation dans laquelle se produisent leurs interactions, au sens large — contexte social, contexte géographique, etc. — et de voir l'interaction comme un phénomène situé. Considérer ces contextes amène à mieux comprendre le fonctionnement des systèmes interactifs et à concevoir des interfaces adaptées ou adaptables aux contextes d'utilisation[5]. En effet, la modélisation de l'interaction uniquement en termes d'utilisateur, de système et de tâches ne correspond pas à certaines applications, comme par exemple les activités créatives et artistiques, où on est souvent amené à détourner les objets et les systèmes des utilisations qui étaient prévues par les concepteurs.

Dourish explique comment le phénomène d'*embodiment*, issu de la phénoménologie — le fait pour quelque chose d'exister à travers notre expérience du monde, de s'y manifester et d'y participer, ce qui inclut la manifestation physique dans le monde sensible — amène à changer notre point de vue sur l'interaction tangible[27]. Dans un hôpital, les fiches qui décrivent l'identité et les traitements administrés aux patients ne représentent pas seulement les informations qui y sont écrites, elles portent également la trace de la manifestation dans ce monde des activités dans lesquelles ces fiches interviennent : l'écriture indique l'auteur de la fiche, ou son usure renseigne sur la fréquence d'utilisation. Ces informations sont exploitées par les personnes pour qui leur manifestation physique a un sens. On voit donc que l'interaction avec les objets dans le monde dépasse le cadre pour lequel ils ont été conçus, que les représentations et les significations qui sont manipulées à travers les objets sont subjectives et dépendent de l'activité, que ce sont elles qu'on manipule à travers les objets[26], et que l'*embodiment* caractérisant les interfaces tangibles peut laisser envisager de nouvelles perspectives pour leur conception.

En conséquence, la vision de l'interaction comme un traitement de l'information en permanence ne permet pas de l'expliquer entièrement. Cette vision cartésienne et objective laisse place peu à peu à une vision tournée sur l'activité et la pratique. Ce changement de vision s'effectuerait sur quatre tableaux[30]. Au lieu de regarder les objets tangibles comme des porteurs d'une représentation dont le sens est défini par le concepteur du système, il est possible de les voir comme des moyens de réaliser plusieurs types d'action[29] : de réaliser des actions, par l'intermédiaire du virtuel, qui seraient impossibles à réaliser dans le vie réelle ; d'augmenter la perception ; de rendre possibles la manipulation physique ; ou bien



**FIGURE 2.2** Modèle basique de l'interaction musicale

de réaliser des actions référentielles, sociales ou contextuelles, comme partager un objet. Au lieu de se considérer que seules les fonctionnalités du système définissent l'interaction entre le système et son utilisateur, la conception doit mettre en avant ce que les utilisateurs pourront faire dans les contextes dans lesquels ce système va être utilisé. Au lieu de concevoir le système tourné vers un seul utilisateur, il faut pouvoir envisager que son utilisation peut être partagée. Enfin, au lieu d'imposer une signification particulière, la conception du système doit permettre à chacun d'avoir sa propre interprétation de son utilisation en favorisant les utilisations non prévues.

## 6. Conclusion

Nous avons présenté les enjeux de l'interaction homme-machine à travers quelques unes de ses évolutions. Nous avons ensuite détaillé comment plusieurs styles d'interaction se sont retrouvés autour du même objectif, intégrer l'informatique dans le monde réel. Enfin, nous avons vu de quelle manière un style d'interaction particulier, l'interaction tangible, permettrait de répondre de manière pertinente à cet objectif, à travers une vision située de l'interaction.

## 2 Interaction tangible pour la musique

L'informatique musicale est la discipline qui étudie les relations entre l'informatique et la musique. Ces relations sont complexes et couvrent une quantité de problèmes, mais nous nous intéressons ici à l'étude des systèmes interactifs de production sonore dans le cadre de la performance musicale. Nous verrons que la production de sons par l'ordinateur a des implications particulières qui ne facilitent pas la performance musicale de la manière dont on utilise des instruments acoustiques traditionnels, et que l'interaction musicale soulève notamment la question de la réalisation du contrôle musical. Nous montrerons enfin les avantages de l'interaction tangible pour ce contrôle.

### 1. Interaction musicale

Traditionnellement, la production de la musique se fait par l'interaction entre le musicien et l'instrument, par l'intermédiaire du geste. La production de musique avec un ordinateur n'obéit pas à cette règle : un clic énergique sur une souris peut provoquer la production, ou *synthèse*, d'un son quelconque. On peut également planifier une série d'événements sonores en laissant à l'ordinateur la tâche d'assurer l'exécution en temps de ces événements. Ainsi, de nombreuses activités musicales peuvent être envisagées grâce à l'ordinateur et à l'immense richesse sonore qu'il peut produire. Il est possible de préparer ou de programmer une séquence musicale d'une complexité arbitraire en paramétrant n'importe quel aspect. L'ordinateur peut générer toute la musique qui est spécifiée à l'avance. Comme nous pouvons distinguer, dans un concert, autant de sources sonores qu'il y a d'instruments, nous

pourrons distinguer plusieurs sources sonores produisant différents sons de manière concurrente lorsque l'ordinateur produit la musique. Chacune de ces sources sera appelée *processus sonore*, ou *processus musical*. La distinction des processus musicaux est subjective et peut être faite en fonction de la perception — on se représente les sons perçus comme différents — ou bien de la cognition — on se représente les structures qui produisent le son et quelles sont leurs relations telles qu'on les a conçues.

L'interactivité en informatique musicale correspond à l'activité de performance musicale. Cette activité est traditionnellement incarnée par le jeu instrumental, dans lequel un musicien spécifie les modalités de la production sonore à travers des gestes d'excitation, de modification ou de sélection, et l'énergie qu'il transmet dans ces gestes[14]. Or, lorsque l'ordinateur assure la production de la musique, comment spécifier en temps réel ces modalités ? Une des préoccupations de l'interaction musicale est de permettre le *contrôle* interactif des processus sonores.

Les instruments traditionnels possèdent des avantages : ils permettent un contrôle expressif, autorisent la virtuosité et présentent diverses formes. En revanche, ils ont également des inconvénients : leur construction limite l'étendue des sons qu'ils peuvent produire et les formes qu'ils peuvent avoir, et le lien entre le geste et le son qui existe dans ces instruments rend difficile leur utilisation expressive par des musiciens peu expérimentés. L'informatique promet de dépasser ces inconvénients, mais il est difficile d'obtenir les mêmes avantages. C'est là que se trouvent les enjeux du contrôle interactif des processus sonores.

#### FIGURE 2.2

Nous avons vu que l'ordinateur était capable de produire le son en circuit fermé mais également en temps réel. Nous définirons les *contrôleurs* comme les dispositifs qui concourent à la spécification des modalités de la production du son par l'ordinateur. On peut ainsi modéliser le contrôle interactif de façon basique comme une interaction entre trois parties (fig. 2.2) : le *musicien*, les *contrôleurs* et les *processus musicaux*. Le musicien agit, par son système moteur, sur les contrôleurs et la perception qu'il a de ces derniers le guident dans son action. Les contrôleurs ont une action sur les processus musicaux, qui peuvent à leur tour influencer les possibilités de contrôle. Enfin, les processus musicaux produisent ce qui est perçu par le musicien, qui peut en retour agir sur la musique en changeant ses intentions. Ce modèle met bien en avant la particularité du système interactif musical : les contrôleurs et les processus musicaux sont disjoints, à l'inverse de l'instrument pour lequel ces deux parties sont indissociables. Il montre aussi que l'interaction musicale est multimodale : le musicien et son instrument de musique numérique communiquent par le toucher, l'image et le son.

Ce modèle nous sert pour décrire les problématiques du contrôle dans l'interaction musicale et les solutions mises en œuvre. Dans ce modèle, chaque interaction suggère une problématique particulière qui touche le contrôle dans son ensemble :

- l'interaction entre le musicien et les contrôleurs demande une réflexion sur les interfaces qui seront conçues et les paradigmes d'interaction qu'elles supportent ;
- l'interaction entre les contrôleurs et les processus musicaux demande une réflexion sur les mappings, c'est-à-dire les liaisons entre les paramètres de contrôle et les paramètres de synthèse ;
- l'interaction entre les processus musicaux et le musicien demande une réflexion sur les structures et les algorithmes de synthèse.

## 2. Environnements graphiques pour la musique

La première des trois problématiques précédentes nous intéresse ici. Considérons la pratique instrumentale traditionnelle. Les musiciens, après une période d'apprentissage de l'instrument, ne s'appuient plus sur leur vision pour jouer, mais sur leur ouïe et leur toucher. Par

exemple, elle peut servir pour surveiller la communication faciale de ses partenaires lors d'une improvisation de jazz, ou pour lire une partition. À l'inverse, les interfaces graphiques utilisent beaucoup d'images mais très peu de son. La complémentarité de ces deux situations laisse penser que le contrôle interactif des processus musicaux peut se manifester à travers une interface graphique[21].

La représentation visuelle permet d'organiser et de visualiser un nombre très important de paramètres. Cependant, le multiplexage temporel de l'entrée dans les interfaces graphiques limite le nombre de paramètres pouvant être manipulés de manière concurrente. Même en adjoignant à l'interface graphique des contrôleurs physiques externes, ce qui fait perdre le contrôle organisationnel et le sens qu'apportait la représentation graphique, ce qui manipule (la souris ou le contrôleur externe) est dissocié de ce qui est manipulé (le processus sonore qui est représenté graphiquement à l'écran). Plusieurs styles d'interaction peuvent permettre de continuer dans cette voie en évitant les problèmes sus-cités. L'interaction tactile permet de manipuler naturellement les représentations graphiques en deux dimensions avec les doigts sur une surface multipoint. La réalité virtuelle va encore plus loin en permettant d'interagir en immersion avec les représentations graphiques en trois dimensions[9].

### 3. Interaction tangible et musique

L'interaction tangible fait partie des choix pertinents pour réaliser des interfaces musicales[22]. Plusieurs approches ont été envisagées : des systèmes d'objets spatiaux et relationnels sur table[48, 67, 69], des systèmes d'objets constructifs[61, 32], des objets augmentés autonomes[83, 70], etc.

Les aspects de l'interaction tangible liés à la manipulation directe, que nous avons déjà évoqué, sont bénéfiques pour le contrôle musical. Les processus sonores sont représentés non plus graphiquement, mais physiquement, et la coïncidence de leur représentation et des moyens de leur contrôle dans des objets plus spécifiques que génériques ne fait qu'accentuer le réalisme de leur utilisation. On peut envisager qu'il est possible d'interagir avec des interfaces tangibles d'une manière plus proche des instruments traditionnels. Cependant, le potentiel des interfaces tangibles musicales ne s'arrête pas là, car elles permettent aussi de contrôler des structures de plus haut niveau, s'éloignant ainsi du cadre instrumental et se rapprochant du cadre compositionnel[47].

Les interfaces tangibles musicales semblent également profiter à l'interaction homme-machine. Selon Jordà, ces interfaces font des cas d'utilisation intéressants car ils figurent le contrôle précis d'un espace de paramètres continu à plusieurs dimensions, des niveaux d'utilisation variés (débutants et experts, adultes et enfants), une dimension sociale (les objets sont partageables), et enfin l'importance de la créativité et de l'expressivité[47].

### 4. Conclusion

Nous avons défini le contexte de l'interaction dans l'informatique musicale, et avons ensuite découvert ses enjeux à travers un modèle simple qui a permis de dévoiler plusieurs problématiques dans ce contexte. Nous avons enfin vu comment l'une d'elle, la conception de l'interaction musicale, trouve des réponses dans les interfaces tangibles.

Notre objet d'étude, l'interaction tangible pour la construction opportuniste musicale, demande que nous précisions les notions de construction musicale et d'opportunisme ainsi que leurs relations. Nous pourrions ensuite exprimer et détailler la problématique centrale de notre travail.

## 1 Construction musicale dans les interfaces tangibles

Une partie de l'intérêt de l'interaction tangible dans le cadre de l'expression musicale est fait par la possibilité de manipuler des structures complexes de manière réaliste. Cependant, on insiste assez peu sur les possibilités de construction qu'offrent les interfaces tangibles, et peu d'entre elles peuvent réellement dispenser le musicien d'utiliser une interface plus traditionnelle pour préparer sa performance. Nous décrivons l'activité de construction dans la performance musicale et tenterons de donner des pistes pour caractériser cette activité dans les interfaces tangibles.

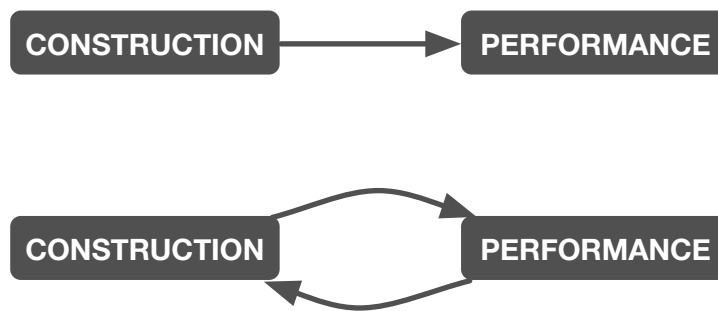
### 1. Construction et performance musicale

Tout au long de ce document, on désignera par construction l'activité qui consiste à mettre en place la structure de production sonore dans le cadre de la performance musicale. Cette définition englobe une grande variété d'actions. Il s'agit de la mise en place des contrôleurs, des processus musicaux, des mappings entre eux, des instruments, des haut-parleurs ou des éventuelles liaisons entre ces éléments.

#### FIGURE 3.1

Si on analyse une situation de jeu musical acoustique, on observe généralement deux phases : la préparation de l'instrument et le jeu instrumental. Au cours de la préparation, le joueur de guitare classique s'assure du bon état de ses cordes et accorde son instrument. Puis le jeu instrumental se traduit par des gestes contraints par les propriétés physiques de l'instrument, qui définissent un cadre d'action figé dans lequel le musicien peut évoluer. L'introduction de l'outil informatique dans le processus de performance musicale a bouleversé ces phases en offrant au musicien la possibilité d'imaginer et de construire de nouvelles sources sonores et de nouveaux moyens de contrôle. Malgré cette liberté, les procédés de création musicale mis en œuvre à travers ces nouveaux outils sont majoritairement influencés par le jeu musical acoustique, et la construction reste bien distincte du jeu musical. Les deux activités s'articulent autour d'un processus linéaire : la construction de la structure de production sonore (*construction en temps différé*) précède son utilisation. Pourtant, il est également possible d'envisager un schéma plus libre (*construction en temps réel*, ou *reconfiguration dynamique*<sup>1</sup>), où il serait possible de modifier la construction sans interrompre

<sup>1</sup> Dans la suite de ce document, on utilisera le terme *reconfiguration dynamique*, plus proche du terme anglais de *dynamic patching*.



**FIGURE 3.1** Construction en temps différé (en haut) et reconfiguration dynamique (en bas).

la performance. Ces deux manières d'aborder la construction sont représentées sur la figure 3.1.

Pour la musique acoustique, la reconfiguration dynamique reviendrait par exemple à modifier un instrument (par rapport à son fonctionnement habituel) dans le même temps qu'il est joué. Lorsque le son est représenté sous forme de signal, on peut alors dire que cela reviendrait à en changer le trajet. L'utilisation d'une *no-input mixing board* de Toshimaru Nakamura, une table de mixage utilisée comme instrument en connectant les entrées aux sorties et en jouant sur l'effet Larsen produit, constitue un exemple de reconfiguration dynamique. Dans le domaine de l'informatique musicale, la modification des mappings entre des contrôleurs et des processus sonores rentre également dans le cadre de la reconfiguration dynamique.

On peut considérer que les notions de construction en temps différé et de reconfiguration dynamique sont liées respectivement à celles de temps différé et de temps réel. Dans ce cadre, il serait stérile de vouloir les comparer. En effet, chacune est associée à des enjeux bien particuliers mais elles sont inextricables. Les travaux de Barkati[3] suggèrent que le temps réel est le monde de l'improvisation, de la performance et de l'éphémère, où l'instantanéité s'obtient au prix de l'irréversibilité, tandis que le temps différé est celui de la composition, de la partition et de la pérennité, où la distanciation temporelle rend possible l'édition, le retour compositionnel. Il montre que la séparation stricte entre ces deux univers n'est pas pertinente, et que les pratiques musicales actuelles rendent de plus en plus interdépendants. On peut en conclure qu'il est souhaitable de proposer des systèmes qui rendent possible la reconfiguration dynamique, en plus de la construction en temps différé.

## 2. Utilisations de la reconfiguration dynamique

Dans le schéma de la reconfiguration dynamique, non seulement la musique peut être improvisée, mais les structures qui la produisent peuvent l'être aussi. Il semble même intéressant de considérer la construction comme une partie du jeu musical. La liberté de modifier les structures et le fait d'en apprécier les résultats instantanément font qu'on peut également voir la construction en temps réel comme une opportunité pour le prototypage et l'expérimentation de structures de production sonore, de contrôleurs ou d'instruments de musique numérique. Nous présentons maintenant quelques exemples de la place qu'a actuellement la reconfiguration dynamique dans différents systèmes interactifs musicaux.

Le *live coding* correspond à l'idée d'écrire des programmes informatiques qui sont interprétés en temps réel dans des langages qui sont utilisés pour synthétiser du son ou des images. Le pouvoir expressif et la nature dynamique d'un tel langage autorise toutes les modifications imaginables[19], et la possibilité de construire et de maîtriser des abstractions de haut niveau laisse penser qu'il est possible de maîtriser l'aspect temporel de la reconfiguration dynamique

(*ie.* faire les bonnes reconfigurations au bon moment), d'autant plus que certains langages comme *Chuck*[80] intègrent le temps de manière à pouvoir le contrôler précisément. Il est souhaitable de profiter de tout ce potentiel avec des interfaces plus directes comme celles les interfaces graphiques.

Certains logiciels présentant une interface graphique conventionnelle permettent la reconfiguration dynamique. Les environnements graphiques modulaires qui descendent de *Patcher* (*Max*, *Puredata*) autorisent naturellement cette possibilité, puisque l'utilisateur peut modifier le *patch* qui est créé et qui spécifie le programme de traitement des événements et du signal exécuté en temps réel. Dans une moindre mesure, on peut aussi évoquer la station de travail audionumérique *Live* qui donne la possibilité de rajouter ou de supprimer des instruments ou des effets sans que la musique soit arrêtée. Les reproches qu'on a pu formuler à l'encontre des interfaces graphiques pour le contrôle des sons musicaux vaudraient aussi pour la reconfiguration dynamique. La manipulation des objets à la souris ou au clavier est indirecte et ne facilite pas les opérations de réorganisation qui peuvent être parfois complexes (remplacer *effet1* par un *effet2* implique de débrancher *effet1*, l'isoler, choisir un *effet2*, le placer et le brancher).

FIGURE 3.2

Les styles d'interaction favorisant la manipulation directe semblent plus propices à la reconfiguration dynamique, car, au contraire de certaines solutions qui obligent à dissocier les outils pour la construction et ceux pour le jeu musical, ils rendent accessibles ces deux activités sous les mêmes modalités. L'interaction dite tactile et gestuelle présente l'avantage de réunir sur la même surface, un écran tactile, des possibilités de contrôle et de réorganisation particulièrement riches[25], grâce à reconnaissance des gestes et à la prise en compte de plusieurs points de pression. L'interaction tangible semble une voie également prometteuse. Les interfaces tangibles basées sur les systèmes d'objets se présentent comme de bons candidats pour la reconfiguration dynamique, et c'est une thématique qui commence à émerger à travers certains cas d'utilisation. Les concepteurs de la *reacTable*, une interface tangible sur table où la synthèse modulaire est représentée par la mise en place et la manipulation d'objets sur la table (fig. 3.2), considèrent que la reconfiguration dynamique, qu'ils nomment *dynamic patching*, représente un aspect fondamental de leur instrument[51]. Dans ce système, c'est la configuration spatiale des éléments qui détermine leurs relations, et c'est en changeant les positions des objets sur la table qu'il est possible de reconfigurer dynamiquement les processus sonores en jeu. D'autres systèmes privilégient plutôt l'assemblage physique d'éléments de manière à former des structures de contrôle ou de synthèse, comme *Pin&Play&Perform*[12].

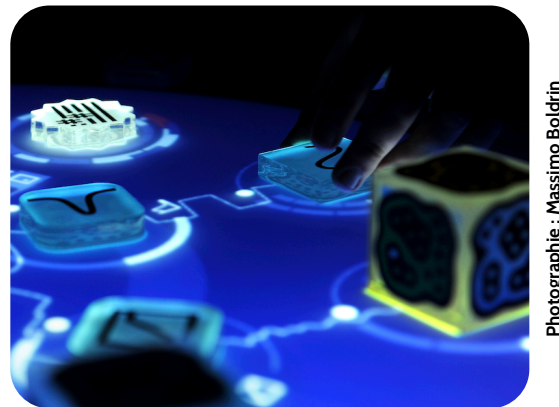
Nous avons donc pu constater les avantages des styles d'interaction favorisant la reconfiguration dynamique, et notamment l'interaction tangible, pour intégrer cette activité sous les mêmes modalités d'action que celle du jeu musical.

### 3. Construction musicale tangible

Nous nous intéressons ici à la reconfiguration dynamique au sein des interfaces tangibles musicales. Afin de guider la conception de telles interfaces, nous proposerons des propriétés qui font que des interfaces permettent la reconfiguration dynamique, et nous évoquerons enfin les limites de l'interaction tangible dans le cadre de la construction musicale.

#### a) Caractérisation

La majorité des interfaces tangibles musicales qui permettent de manipuler des structures de haut niveau proposent des systèmes d'objets qu'il est possible de placer, d'enlever, de combiner. On définit la *composition dynamique* comme la propriété d'une interface tangible à rendre possible la variation arbitraire du nombre d'objets impliqués dans l'interaction. La



Photographie : Massimo Boldrin

**FIGURE 3.2** La reactTable[47] repose sur la reconfiguration dynamique.

composition de l'interface avec des objets, à la manière de blocs de construction, est similaire à l'expression d'un langage, qui permet d'exprimer des concepts, des actions ou des structures complexes.

En étudiant le couplage entre représentation tangible et représentation numérique, Ullmer et Ishii ont distingué deux méthodes de liaison entre les objets physiques et leurs contreparties virtuelles[78] : statique, lorsque la liaison est choisie à l'avance et qu'il n'est pas possible de la modifier, et dynamique, lorsque l'utilisateur peut décider des liaisons. On peut interpréter la liaison comme un changement du comportement de l'objet, de sa réponse aux manipulations extérieures. La *liaison dynamique* sera définie comme la propriété d'une interface tangible à rendre possible un changement arbitraire dans son comportement. Cela invite l'utilisateur à explorer de nouvelles configurations avec les objets déjà utilisés et permet d'envisager des représentations ou des comportements flexibles et dynamiques fusionnables, donnant naissance à de nouveaux types d'objets qui peuvent augmenter le pouvoir expressif.

La reactTable\*[48] permet la composition dynamique car le système permet d'ajouter ou d'enlever des objets sur la table, mais ne possède aucun mécanisme qui rende possible la liaison dynamique. À l'inverse, le Sound of Touch[57], une interface tangible permettant de manipuler une texture sonore de manière expressive grâce à un outil spécialisé, ne permet pas la composition dynamique, car on n'agit que dans un cadre fixé et non extensible, mais il permet la liaison dynamique, car la texture qui est représentée par l'outil spécialisé peut être changée. Il semble donc que ces deux propriétés, la composition dynamique et la liaison dynamique, concourent de manière orthogonale à augmenter le potentiel de reconfiguration dynamique.

### **b** Limites

Envisager la construction par l'intermédiaire de l'informatique tangible nécessite de considérer deux facteurs limitants.

Premièrement, si l'informatique tangible est associée à une forme de persistance, du fait de la présence d'objets physiques qui conservent leur existence même une fois que la partie numérique du système disparaît[43], elle n'en garde pas moins un aspect éphémère, car les opérations de sauvegarde et de chargement si courantes dans le monde virtuel n'ont pas cours dans le monde réel. À moins d'avoir beaucoup d'espace et plusieurs copies d'un système tangible, la persistance des réalisations n'est pas évidente, bien qu'il soit possible d'imaginer des solutions partielles à ce problème. Par exemple, les concepteurs de la Music Table uti-

lisent des objets spéciaux, les *phrase cards*, qui peuvent accueillir et réunir les informations relatives à une structure construite à partir d'autres objets. Même si ces objets sont retirés plus tard, il est toujours possible d'éditer la structure contenue dans les *phrase cards*, dont seule la représentation virtuelle peut être affichée[8].

Deuxièmement, la possibilité de corriger les erreurs qui sont faites, qui est un aspect non négligeable de l'interaction homme machine, et qui est souvent représentée par les simples actions *Annuler* et *Refaire* dans les interfaces graphiques, est moins triviale lorsqu'on interagit avec des objets physiques[6]. Lorsque seule la position des objets importe, l'utilisateur peut corriger ses erreurs en remplaçant les objets, mais lorsque l'état du système est plus dynamique et que des actions sont irréversibles, il est aussi difficile de revenir à la configuration précédente que de séparer l'eau chaude de l'eau froide dans un verre d'eau tiède.

Ces deux problèmes touchent à des fonctions très utiles pour l'édition si caractéristique du temps différé, ils pourraient donc compromettre cet aspect de l'activité de construction. Ces limites ne doivent pas être oubliées dans la conception des interfaces tangibles musicales.

## 2 Opportunisme

Nous proposerons de considérer l'opportunisme, une propriété des systèmes interactifs dont nous montrerons les avantages et les inconvénients. Nous expliquerons ensuite comment elle peut servir à encourager la reconfiguration dynamique dans le cadre des interfaces musicales tangibles.

### 1. Définition

Une des motivations des interfaces tangibles est d'intégrer le calcul (au sens *computation*.) dans le monde réel. Cela suppose donc soit de concevoir des objets physiques spécifiques dotés de capacités de calcul, soit d'augmenter de manière superficielle des objets non spécifiques pour leur apporter ces capacités. Il est important cependant de noter que l'on devrait considérer la modification profonde et durable d'un objet existant comme la conception d'un objet spécifique. Par exemple, un ours en peluche dont le ventre aura été évidé pour accueillir un circuit électronique est clairement considéré comme un objet spécifique. Au contraire, on considérera plutôt qu'un ours en peluche est augmenté lorsqu'on lui attache une puce ou un marqueur visuel, qui peuvent être retirés plus tard sans altérer l'intégrité physique de l'objet. Les objets nous renseignent sur leurs utilisation par leurs *affordances*, c'est-à-dire les possibilités d'utilisation qu'ils nous suggèrent, et leurs *contraintes*, c'est-à-dire les parties qui restreignent le nombre d'actions possibles, nous renseignent sur ce qu'il est possible de faire avec eux[63]. À ces possibilités s'ajoutent celles de la partie augmentée des objets, qui peut également suggérer d'autres affordances ou contraintes.

On définira l'*opportunisme* comme la propriété d'une interface à faire appel à des éléments non spécifiques et inconnus au préalable pour supporter l'interaction.

### 2. Positionnement

L'opportunisme est seulement un nouveau nom donné à une caractéristique présente dans de nombreux systèmes interactifs. Cette caractéristique a déjà été évoquée auparavant mais rentrait dans la description de concepts moins précis. Nous précisons donc ici notre positionnement par rapport aux définitions existantes.

Certains travaux en rapport avec la réalité augmentée peuvent être qualifiés d'opportunistes. C'est le cas, par exemple, du système *Tiles*[72], qui propose de réunir la réalité aug-

	opportunisme	liaison dynamique	composition dynamique
Digital Photo Browser [79]	partiel	oui	oui
memolCON [16]	partiel	oui	oui
Musique opportuniste [39]	total	oui	non
Opportunistic controls [40]	total	non	non
Pin&Play&Perform [12]	non	non	oui
reactTable* [48]	non	non	oui
Surface particles [46]	total	oui	oui
Tactus [56]	total	oui	oui

**FIGURE 3.3** Quelques systèmes interactifs passés au crible des notions que nous avons définies.

mentée et l'interaction tangible en basant l'interaction sur la manipulation de marqueurs visuels pour tirer parti du meilleur des deux mondes. Cette idée a fait naître le terme de *tangible augmented reality*, qui est défini et détaillé dans [10]. Bien que ces recherches puissent fournir une bonne base de travail, nous préférons nous en éloigner pour l'instant et utiliser le terme plus précis d'*opportuniste*, qui caractérise uniquement la propriété décrite dans les paragraphes précédents et met l'accent sur l'utilisation des objets physiques et moins de marqueurs visuels servant de socle à des objets virtuels rendus en trois dimensions.

Le terme *opportuniste* a été utilisé par Henderson et Feiner pour décrire des interfaces qui augmentent des objets inutilisés de l'environnement physique aux affordances physiques intéressantes de manière à les transformer en éléments d'interaction caractéristiques des interfaces graphiques, des *widgets* appelés *opportunistic controls*, dans le cadre d'une tâche de maintenance industrielle[40]. D'autres domaines d'applications sont ensuite traités afin de trouver quels objets présentent les meilleures affordances selon la tâche et l'élément de contrôle virtuel voulu (bouton, potentiomètre, etc.)[41]. La *musique opportuniste* [39] exploite l'idée des *opportunistic controls* dans le cadre du contrôle musical. Toutefois, l'objectif des *opportunistic controls* nous semble loin de drainer tout le potentiel de l'interaction tangible. En effet, l'accent est porté sur l'exploitation d'objets physiques connus au préalable et de notre capacité à exploiter leurs affordances pour en utiliser certaines parties comme des *widgets*, du bout du doigt, alors que leur préhension et leur manipulation peut suggérer bien d'autres modalités d'interaction. Nous conservons donc ce terme d'*opportuniste* car il correspond bien à l'idée qu'il représente, mais souhaitons garder une certaine distance également avec les *opportunistic controls* qui ne représentent qu'une partie cette l'idée.

### 3. Caractérisation

Cette notion permet de caractériser les systèmes existants dont nous détaillons quelques exemples qui sont synthétisés par la figure 3.3. À partir de ces cas, nous pourrions décrire ainsi comment il est possible de parler de l'*opportunisme*.

*Pin&Play&Perform*, une interface tangible que nous avons déjà évoqué à la page 14, ne peut pas être qualifiée d'opportuniste, car tout objet impliqué dans l'interaction avec ce système est spécifique. Il en va de même pour la *reactTable\**, bien qu'il ne soit pas impossible d'imaginer superposer des objets de la vie courante par-dessus les marqueurs. À l'inverse, les *Opportunistic controls* et la *Musique opportuniste* peuvent être considérés comme totalement opportunistes puisque l'interaction s'appuie presque entièrement sur l'utilisation d'objets environnants non spécifique. Le séquenceur *Tactus*[56] permet d'utiliser n'importe quel objet ou dessin représentant une grille remplie ou non d'objets de couleurs comme interface pour le séquençage musical, à la manière des step-séquenceurs logiciels. Là aussi, l'interac-

tion est totalement opportuniste, car entièrement portée par des objets non spécifiques (un jeu de Puissance 4 ou une feuille de papier, par exemple). Les *surface particles* sont des éléments d'interaction combinant un affichage sur une surface quelconque par un projecteur et l'utilisation d'un stylet infrarouge[46]. Les objets de l'environnement immédiat deviennent des surfaces interactives qui supportent la quasi-totalité de l'interaction.

Certains systèmes se prêtent mal à une évaluation binaire de l'opportunisme. C'est le cas du *Digital Photo Browser*[79], un système qui permet d'attacher des photographies numériques à des objets non spécifiques, de manière à en faire des objets-souvenirs augmentés. Il est composé d'une table qui permet de reconnaître un objet posé dessus que l'on souhaite augmenter avec des photographies. Cette tâche, par contre, est réalisée grâce à un écran tactile. Ainsi, seule une partie de l'interaction est de nature opportuniste. L'interface tangible sur table *memoICON*[16] propose d'augmenter les tables tactiles en permettant d'utiliser des objets du quotidien comme des icônes. Si ces objets jouent une part dans l'interaction, une grande partie de cette dernière reste tactile et gestuelle. Ces deux exemples suggèrent qu'il serait plus précis de donner une mesure de la notion d'interface opportuniste. On se limitera ici à parler d'interfaces non-opportunistes, d'interfaces partiellement opportunistes ou d'interfaces totalement opportunistes. On s'intéressera plus particulièrement aux interfaces totalement opportunistes dans la suite de notre étude.

## 4. Vers une construction musicale opportuniste

La définition que nous avons donné d'une interface opportuniste fait apparaître des liens avec les deux caractéristiques que nous avons évoqué plus haut, la composition dynamique et la liaison dynamique. Une interface permettant d'utiliser les objets physiques de l'environnement immédiat comme médiateurs de l'interaction pose deux questions importantes. La première est : *Quels objets impliquer ?* La réponse peut être trouvée à l'avance, mais aussi en plein milieu de l'utilisation, et ce cas entre dans le cadre de la composition dynamique. La seconde question est : *Comment impliquer ces objets alors qu'on ne les connaît pas a priori ?* Là aussi, la réponse peut être trouvée à l'avance (par exemple, ils seront tous utilisés de la même manière précise), ou bien pendant l'interaction, ce qui rentre dans le cadre de la liaison dynamique. Ainsi, l'opportunisme peut faire apparaître naturellement ces caractéristiques qui sont recherchées pour la reconfiguration dynamique. Cependant, il faut prendre en compte ses avantages et ses inconvénients pour orienter la conception.

### a) Avantages

Le choix de concevoir une interface opportuniste peut relever du type d'application ou des contraintes d'un projet particulier, mais il peut être orienté par certains avantages qui caractérisent cette approche. Nous en avons retenu quatre qui pouvaient toucher la construction musicale opportuniste : l'accessibilité, la personnalisation, l'utilisation de l'espace et la mobilité.

**Accessibilité** Les interfaces opportunistes réduisent le *coût d'entrée* pour l'utilisation d'un système. En effet, puisque l'interaction repose uniquement sur les objets qui entourent l'utilisateur, le système doit pouvoir supporter du matériel générique pour fonctionner. Dans la pratique, le matériel nécessaire est assez peu coûteux : caméras, marqueurs imprimés sur papier, puces RFID, etc.

**Personnalisation de l'interaction** On a vu que les recherches récentes en interaction insistent sur l'importance de prendre en compte le contexte d'utilisation et le modèle mental

des utilisateurs dans la conception des systèmes interactifs. Les interfaces opportunistes sont en quelque sorte un moyen de laisser l'utilisateur faire une partie de ce travail de conception. Les affordances et les contraintes qu'il perçoit des objets, ses goûts artistiques, sa créativité, ainsi que sa propre expérience avec ceux-ci, peuvent le guider pour définir des utilisations pertinentes dans le cadre d'une interaction avec le système.

**Utilisation de l'espace et mobilité** En général, les interfaces tangibles situent l'interaction dans un espace limité, le *cadre de référence*. Cela peut être, par exemple, la table en dehors de laquelle les objets de la *reactTable* ne sont que des morceaux de plastique dépourvus des effets qu'ils ont lorsqu'ils sont sur la table. Cela constitue une discontinuité spatiale[72] qui *casse l'immersion du réel dans le virtuel*. Une interface opportuniste peut être mobile et même multi-utilisateurs, ce qui rétablit cette *immersion*.

L'utilisation d'un espace potentiellement infini permet également d'envisager une complexité importante dans la construction musicale. En effet, cela permet de répartir les objets en plusieurs endroits, par exemple pour contrôler plusieurs aspects d'un même instrument à différents endroits. Un espace de manipulation potentiellement infini permettrait également aux utilisateurs de s'appuyer plus souvent sur leurs actions épistémiques.

#### **b** Inconvénients

L'opportunisme n'est cependant pas la réponse à tout, et en dehors des nombreux problèmes pratiques que nous aurons l'occasion de rencontrer, certains problèmes théoriques doivent être considérés comme autant de prix à payer pour décider si la conception d'une interface a besoin d'être opportuniste ou pas : le fait que l'opportunisme implique une forme de généralité, une interaction subtile plus difficilement atteignable, et une prise en main potentiellement complexe.

**Spécificité contre généralité** Une des caractéristiques des interfaces tangibles est de permettre l'utilisation d'objets spécifiques, au même titre que les interfaces graphiques, dont la forme et le comportement sont conçus *sur mesure*. Cet avantage disparaît en partie dans les interfaces opportunistes car le concepteur n'a pas le contrôle de la représentation physique des objets qui seront utilisés, et doit prévoir des représentations ou des comportements génériques.

**Interaction moins subtile** Un corollaire de la remarque ci-dessus est qu'il est moins aisé de concevoir techniquement une interaction subtile. En effet, lorsque les objets sont spécifiques, comme dans le cas de l'interaction tangible traditionnelle, il est possible de réunir autant de moyens que possible pour créer des interfaces qui répondent à des interactions de natures différentes. Par exemple, il est bien plus probable qu'une interface non opportuniste puisse réagir de manière subtile à des effleurements, des frôlements, etc.

**Prise en main** Le fait de laisser à l'utilisateur la responsabilité de définir explicitement l'interaction rend plus complexe le premier contact avec le système et son exploration naturelle.

### **3** Formulation de la problématique

Nous avons identifié les intérêts de la reconfiguration dynamique et de son incarnation dans les interfaces tangibles pour dépasser le cadre traditionnel de la performance musicale.

Nous avons ensuite observé la pertinence l'opportunisme pour développer des interfaces tangibles permettant la reconfiguration dynamique.

Dans ce cadre, que nous appellerons la construction musicale opportuniste, de nombreuses questions ne trouvent pas encore de réponse. Comment établir des liaisons entre des objets physiques inconnus au préalable et des objets virtuels ? Quels indices récupérer, quels capteurs utiliser pour obtenir des informations sur ces objets non spécifiques ? Comment garder une bonne qualité d'interaction avec des objets de nature inconnue au préalable ? Certaines questions vont au-delà du cadre de notre étude et trouvent des éléments de réponse dans d'autres champs d'étude.

La captation est un problème couramment évoqué dans les styles d'interaction dits *post-WIMP*. Nous avons ciblé un contexte précis qui demande de reconsidérer la question de la captation pour les interfaces musicales opportunistes. Il s'agit donc de déterminer quel effet les contraintes de notre contexte peuvent avoir sur le choix et la conception de la captation dans les interfaces opportunistes, et quels problèmes peuvent survenir.

Le lien entre opportunisme et reconfiguration dynamique fait naître le besoin de préciser explicitement si un objet physique de l'environnement immédiat doit jouer un rôle dans le système, et quel rôle il doit jouer. Il s'agit donc de déterminer quelles techniques d'interaction peuvent répondre à ce besoin dans le cadre d'un système opportuniste.

Nous articulons donc notre étude autour de ces deux problématiques :

- les méthodes pour la captation opportuniste ;
- et les techniques d'interaction pour la reconfiguration dynamique opportuniste.

Notre démarche pour traiter ces problématiques est décrite dans le chapitre 4 et nous présenterons une mise en application dans un système qui concrétise notre point de vue au chapitre 5.

Le chapitre précédent nous a permis de définir un ensemble de termes pour détailler une problématique générale. Nous avons ensuite isolé deux aspects importants de cette problématique. Dans ce chapitre, nous détaillerons notre démarche, les outils que nous utilisons et les propositions que nous avons faites pour traiter chacun de ces aspects.

Une grande partie des styles d'interaction basés sur le réel s'appuie sur l'utilisation de capteurs pour percevoir l'environnement physique. Le choix et l'utilisation appropriée de dispositifs de captation est une partie importante du processus de conception de tels systèmes. Par exemple, certaines interfaces tangibles proposent des objets spécialisés, munis de capteurs, qui réagissent de façon proportionnée aux torsions ou aux effleurements de l'utilisateur. Les interfaces opportunistes posent une importante contrainte sur la captation, puisque les objets qui supporteront l'interaction ne seront connus qu'à l'utilisation. Il est donc beaucoup moins aisé de tirer des informations dans ce contexte. Nous présentons donc une étude qui suggère des choix pertinents pour la captation opportuniste. Nous verrons qu'un de ces choix, la vision, possède des avantages non négligeables pour obtenir des informations pertinentes, mais également quelques points faibles, que nous proposons d'atténuer ensuite grâce à deux méthodes : la segmentation pour le suivi de marqueur, et le couplage virtuel. Enfin, nous faisons le point sur les techniques d'interaction opportunistes pour la reconfiguration dynamique.

## 1 Choix des capteurs pour l'interaction opportuniste

Le choix des systèmes de captation repose en partie sur les *capacités perceptives* dont on souhaite doter un système interactif, c'est-à-dire ce qu'on souhaite faire percevoir au système : la présence d'une personne, la parole, le mouvement d'un objet, etc. La conception de systèmes de captation est un domaine complexe qui réunit plusieurs disciplines comme l'électronique, la vision par ordinateur, la reconnaissance de formes ou encore l'apprentissage automatique, et il est en général assez difficile d'extraire des informations fiables de ces systèmes. Heureusement, les progrès dans ce domaine sont rapides. Antifakos et al.[1] suggèrent cependant qu'il existera toujours des problèmes dans ces systèmes et qu'il est important de prendre en compte leurs inconvénients dans la conception de systèmes interactifs. Ils définissent le *système de captation* comme l'association de capteurs et d'algorithmes de reconnaissance et évoquent deux causes de problèmes pour l'interaction : l'incertitude, qui est due à des facteurs comme la robustesse, le taux de reconnaissance, la précision ou l'ambiguïté, et la latence.

Nous discuterons d'abord des contraintes qu'impose le contexte de notre étude sur le choix des systèmes de captation, puis établirons l'importance de la vision, dont nous verrons également les problèmes, ce qui nous amènera à considérer une captation multimodale.

## 1. Contraintes

Nous nous intéressons aux systèmes interactifs opportunistes. Cela implique plusieurs contraintes qui ont un impact direct sur le choix des systèmes de captation.

Premièrement, les objets physiques sur lesquels reposeront l'interaction ne sont pas spécifiques à ces systèmes. En effet, les objets qui seront utilisés n'ont pas été conçus pour être utilisés de cette façon, ils ont chacun des utilités et des raisons d'exister en dehors d'un tel système, et ils doivent pouvoir être utilisés sans nécessiter de modification profonde ou durable. Les systèmes de captation utilisés doivent alors être *externes*, c'est-à-dire qu'ils doivent nécessiter au pire une instrumentation légère des objets à considérer, comme par exemple la mise en place d'un marqueur visuel ou d'un *tag RFID*, et au mieux aucune instrumentation (contrainte d'*externalité*).

Deuxièmement, les objets physiques sur lesquels reposeront l'interaction sont choisis à l'utilisation. Contraindre l'apparence physique des objets qu'il est possible d'utiliser revient à être moins *opportuniste*. Il est donc important de réaliser le moins de présomptions possibles sur l'apparence physique ou la forme des objets (contrainte de *généricité*).

Troisièmement, les objets physiques sur lesquels reposeront l'interaction font en principe partie de l'environnement immédiat, qui sera déterminé également à l'utilisation. Contraindre l'environnement en fixant un cadre de référence ou une installation particulière — par exemple, une matrice de microphones permettant de détecter l'activité en trois dimensions autour d'une surface donnée[77] — revient à être moins *opportuniste*. Il est donc important, là aussi, de présumer le moins possible de l'environnement dans lequel l'interaction avec le système aura lieu (contrainte de *mobilité*).

Quatrièmement, l'interaction opportuniste repose sur les objets physiques choisis, et non sur les actions d'un utilisateur à l'égard de ces objets. Favoriser ce point de vue renforce l'aspect *tangible* de l'interaction et correspond à notre vision de l'opportunisme, car l'interaction n'est pas limitée à un utilisateur mais peut faire partie d'un processus collaboratif. De plus, cela offre la possibilité d'utiliser des objets mécanisés, comme une voiture télécommandée, ou bien des causes extérieures, comme une chute de dominos, pour exprimer des actions. Il est donc important d'appuyer la captation le moins possible sur les actions de l'utilisateur, et le plus possible sur les objets physiques impliqués (contrainte de *non-subjectivité*).

## 2. Importance de la vision

Les quatre contraintes qui ont été évoquées restreignent les choix possibles pour les systèmes de captation. Il faut donc privilégier les capteurs génériques, qui ne nécessitent pas une instrumentation importante ni des objets physiques à percevoir, ni de l'environnement, et qui ne sont pas dépendants des actions de l'utilisateur. Cela restreint une bonne partie des systèmes de captation que l'on peut rencontrer dans les interfaces tangibles. Nous prenons à présent des exemples qui suggèrent la difficulté de respecter toutes les contraintes énoncées.

La *Sensetable* est un système de suivi d'objets spécialisés (*pucks*) sur table par électromagnétisme qui permet de récupérer l'identité, la position, l'orientation et l'état des objets suivis[66]. Ce type de système de captation ne satisfait pas la contrainte d'*externalité*, puisque les *pucks* sont munis d'un circuit électronique interne qui contrôle une émission radio dont la fréquence est déterminée par le contact avec l'objet, ni la contrainte de *mobilité*, puisque le système est limité à une table bien spécifique. Cependant, il respecte la contrainte de *généricité*, puisqu'aucune présomption n'est faite sur les objets, qui peuvent être d'une forme quelconque, et la contrainte de *non-subjectivité*, puisque la captation n'est pas dépendante des actions de l'utilisateur.

Le système *TouchSound* est un bracelet composé de plusieurs types de capteurs qui peuvent percevoir des informations variées sur n'importe quel objet présent dans l'environnement immédiat[70] : pression, température, mouvement, magnétisme, conductivité et couleur. Ce système de captation est porté au bras et ne permet d'extraire des informations que d'un objet à la fois, celui qui est actuellement touché. Il ne satisfait donc pas la contrainte de non-subjectivité. Cependant, il respecte la contrainte d'externalité, puisqu'aucune instrumentation n'est faite sur les objets qui supportent l'interaction, la contrainte de généricité, puisque n'importe quel objet peut être manipulé, et enfin la contrainte de mobilité, puisqu'aucune présomption n'est faite sur l'environnement.

Chez l'être humain, la vision est un sens important car il permet de percevoir, sans contact, de nombreux détails de ce qui se trouve dans un volume donné de l'espace qui nous entoure. L'émission ou la propagation des ondes lumineuses détermine ce que nous percevons le plus souvent des objets physiques. Une seule image peut nous renseigner sur une grande variété de caractéristiques de plus ou moins haut niveau : couleurs, textures, forme, aire, volume, position, orientation, etc. Grâce à la combinaison de caméras et d'algorithmes sophistiqués, la vision artificielle satisfait plutôt bien les quatre contraintes énoncées plus haut. De nouvelles images sont récupérées d'une ou plusieurs caméras chaque intervalle de temps fixé, puis elles sont traitées par des algorithmes qui permettent d'en extraire des informations de haut niveau. La vision est très utilisée dans les applications de réalité augmentée et mérite une place privilégiée parmi les systèmes de captation opportuniste.

On supposera que les interfaces opportunistes supportant la construction musicale s'appuient sur des systèmes d'objets spatiaux ou relationnels, et nécessitent en premier lieu de remplir trois fonctions perceptives en temps réel :

- identifier les objets, c'est-à-dire pouvoir associer à chaque objet physique perceptible une identité qui lui est propre, de manière à pouvoir lui associer des informations de manière durable ;
- suivre les objets, c'est-à-dire pouvoir localiser chacun d'eux au fil du temps ;
- estimer la position et l'orientation, ou *pose*, des objets dans l'espace physique.

Identifier, suivre et estimer la pose de plusieurs objets sont trois objets d'étude bien distincts dont les solutions sont aussi variées que complexes. Des algorithmes efficaces ont été développés pour une grande variété de situations dans les cas où les objets physiques ne sont pas instrumentés[55], mais il est encore difficile de mettre en œuvre une solution s'approchant des besoins de la captation opportuniste. On peut citer le système BLORT, qui repose sur la modélisation des objets en formes géométriques tridimensionnelles simples comme des cubes ou des cylindres[60], ou encore OASIS, qui repose sur une base de données de modèles d'objets de la vie courante[88].

Ces méthodes ne sont pas encore répandues dans les systèmes interactifs tangibles et de réalité augmentée, dont les concepteurs préfèrent encore une méthode plus simple basée sur des marqueurs visuels, ou *fiducial tracking*. Des marqueurs visuels exhibant des formes géométriques particulières en deux dimensions sont déposées sur les objets à prendre en compte dans l'interaction. Ce sont ces marqueurs visuels qui sont identifiés, suivis et dont la pose est estimée : c'est une manière indirecte de suivre des objets. Les avantages sont multiples : les algorithmes sont beaucoup plus simples, robustes et peuvent être implémentés en temps réel sans limite sur le nombre d'objets à considérer en une seule fois. Cela élimine également toutes les ambiguïtés que peuvent présenter les méthodes plus complexes. En effet, il est difficile de définir la notion d'objet tant celle-ci peut varier selon les besoins, et on ne peut pas distinguer un objet uniquement par son apparence, c'est-à-dire ce qu'on peut en voir au cours du temps. Par exemple, un marteau est la réunion d'un manche et d'une tête de marteau, et nous pouvons voir ce marteau tantôt comme un objet constitué de deux

parties, ou bien de deux objets qui sont liés par une contrainte mécanique (aucun degré de liberté).

Cette ambivalence de la notion d'objet nous amène à considérer le suivi par marqueur comme une voie pertinente pour la construction opportuniste, car l'utilisateur a la responsabilité de distinguer les objets selon ses critères, ce qui libère le système de cette tâche, élimine le risque d'interprétation erronée, et permet de s'appuyer sur le modèle mental de l'utilisateur. Cependant, un défaut de cette approche est d'obliger l'instrumentation préalable des objets, ce qui peut ne pas être souhaitable.

### 3. Multimodalité de la captation

Mettre en place une captation opportuniste en s'appuyant uniquement sur des techniques de vision artificielle pose plusieurs problèmes. Le plus gênant est sans nul doute la complexité des algorithmes utilisés, qui peut empêcher d'atteindre une latence assez confortable. Le contrôle musical requiert une latence inférieure à une dizaine de millisecondes et une gigue qui n'excède pas une milliseconde, pour les utilisations les plus expressives[85]. Une latence plus élevée compromet grandement certaines formes de contrôle musical qui demandent une précision temporelle élevée, comme les gestes d'excitation. De plus, il est complexe d'arriver à une précision spatiotemporelle acceptable pour la détection de collisions, par exemple. Dans ces cas, lorsque le contrôle musical doit être précis, un compromis doit être fait pour tendre vers une solution de captation multimodale. Par exemple, le prototype de la *Musique opportuniste* utilise un algorithme de vision artificielle pour repérer la position d'un doigt dans l'espace, mais la pression du doigt est détectée par un capteur FSR pour permettre le déclenchement d'événements musicaux de manière quasiment instantanée, sans avoir à souffrir de la latence induite par les algorithmes de vision.

### 4. Conclusion

Nous avons déterminé que les systèmes interactifs opportunistes posaient quatre contraintes pour la captation : externalité, généricité, mobilité et non-subjectivité. La conception des systèmes de captation opportuniste doit donc chercher à respecter au mieux ces contraintes, ce qui nous a amenés à considérer la vision artificielle comme une solution acceptable dans ce contexte. Nous avons enfin constaté que l'application à la construction et de la performance musicale nécessitait d'envisager une captation multimodale.

## 2 Segmentation pour le suivi par marqueurs

Nous avons vu qu'il est possible de suivre des objets quelconques et les repérer dans l'espace en leur attachant des marqueurs visuels, ce qui constitue une solution simple et commode. Cependant, les algorithmes de suivi par marqueurs ne permettent pas de récupérer plus d'informations à propos de ces objets que leur pose et leur identité. D'autres méthodes doivent alors être employées pour obtenir des informations complémentaires comme la forme, le volume apparent ou la couleur des objets. Nous présentons une méthode simple basée sur la segmentation d'images de profondeur pour compléter les informations des algorithmes de suivi par marqueurs.

## 1. Méthode

Soit  $O$  l'ensemble des objets visibles et détectables avec un suivi par marqueurs. On définit une image  $I$  constituée de pixels. Si l'image  $I$  est issue de la captation, alors on considère l'ensemble fini  $O_V \subset O$  des objets visibles dans l'image  $I$ . La suivi par marqueurs permet de définir une application  $M : O_V \rightarrow I$  qui associe à chaque objet visible dans l'image  $I$  le pixel de l'image correspondant au centre du marqueur qui lui est associé.

La segmentation d'une image  $I$  consiste à construire une partition  $C = \{C_1, \dots, C_n\}$  de  $I$ , telle que :

$$\begin{cases} \forall i \neq j, & C_i \cap C_j = \emptyset \\ \forall i, & C_i \neq \emptyset \\ \bigcup_i C_i = I \end{cases}$$

La manière de construire cet ensemble diffère selon les critères choisis[81]. Les êtres humains sont capables de segmenter une image en objets de manière sémantique, car ils connaissent bien les propriétés des objets qu'ils reconnaissent. La segmentation artificielle peut être également effectuée de cette manière, en utilisant la reconnaissance d'objets, mais il est plus courant d'opérer uniquement à partir d'informations caractéristiques de plus bas niveau fournies par les pixels d'une image.

Dans les images issues de la lumière visible, la segmentation se heurte à des problèmes de perception. Par exemple, une balle rouge sera difficile à distinguer sur un fond rouge. Une alternative intéressante pouvant permettre une segmentation basée sur les formes des objets est d'utiliser des cartes de profondeur, c'est-à-dire des images où chaque pixel correspond à un niveau de profondeur situé dans la plage dynamique  $[\alpha, \beta]$ . Ces cartes de profondeur peuvent être obtenues grâce à un algorithme de vision stéréoscopique[35] ou bien grâce à d'autres procédés.

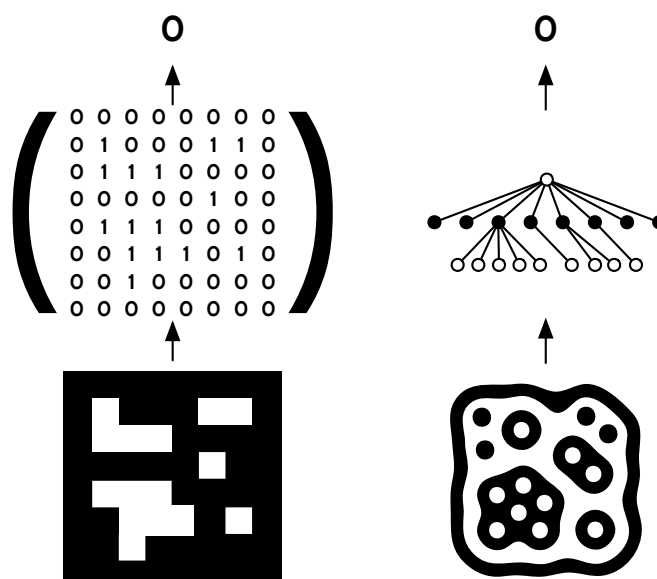
On définit un couple d'images  $(I_c, I_d)$  constitué d'une image en couleurs  $I_c$  et d'une carte de profondeur  $I_d$  de mêmes dimensions et alignées, c'est-à-dire que chaque pixel de coordonnées  $(x, y)$  des deux images représente le même point dans la réalité.

Notre proposition consiste à :

- réaliser un suivi par marqueurs sur  $I_c$  de manière à définir l'application  $M$  ;
- réaliser une segmentation sur  $I_d$  de manière à obtenir l'ensemble  $C$  ;
- définir une application injective  $T : O_V \rightarrow C$  qui associe un objet visible dans le couple  $(I_c, I_d)$  à un segment ;
- exploiter les segments associés aux objets perçus pour obtenir des informations supplémentaires sur ces objets, comme par exemple le volume perçu.

## 2. Suivi par marqueurs

Le principe garantissant le fonctionnement du suivi par marqueurs est de disposer de marqueurs visuels dont les caractéristiques sont parfaitement connues et font qu'il est difficile de trouver de tels motifs dans une image de manière accidentelle. Les formes des marqueurs visuels ont évolué en même temps que les algorithmes de suivi. Au départ, ces marqueurs étaient des suites de barres colorées[76] ou des codes-barres[75] codant des identifiants uniques. Plus tard, des méthodes ont permis d'estimer la pose des marqueurs et ainsi de l'utiliser pour repérer le déplacement de la caméra dans l'espace[53]. Aujourd'hui, plusieurs bibliothèques logicielles comme ARTToolkitPlus[24] ou reacTIVision[49], permettent d'exploiter des méthodes de suivi de plusieurs marqueurs spécifiques dans une image à des fréquences assez élevées pour permettre une utilisation interactive des objets suivis. Les travaux



**FIGURE 4.1** Marqueur matriciel (*ARToolkitPlus*, à gauche) et marqueur topologique (*reactIVision*, à droite).

les plus récents cherchent à améliorer la robustesse de la reconnaissance des marqueurs[7] et la quantité d'information qu'ils peuvent représenter[62].

**FIGURE 4.1**

Ces marqueurs sont réalisés de façon à coder un identifiant et à minimiser le risque d'erreur de reconnaissance dans le cas de mauvaises conditions d'éclairage ou d'occultations partielles. Deux méthodes prédominent aujourd'hui pour la conception des marqueurs (fig. 4.1). Ceux qui reposent sur des matrices à deux dimensions qui exhibent des motifs particuliers (images ou codes) sont identifiés par corrélation avec un dictionnaire de marqueurs connus ou en les transformant en une suite de bits codée par un code correcteur[31]. Ceux qui reposent sur l'adjacence de régions exhibent une structure topologique particulière qui est identifiée en construisant un arbre d'adjacence des régions[20].

Les marqueurs visuels sont presque toujours des images binaires. L'approche classique du suivi de marqueurs consiste à effectuer un seuillage sur l'image d'origine afin de la rendre binaire, puis à détecter les marqueurs sur l'image résultante et enfin à estimer leur pose. Chaque étape laisse le choix de plusieurs algorithmes adaptés. Nous avons utilisé ceux des bibliothèques logicielles *ARToolkitPlus*[24] et *reactIVision*[49].

### 3. Segmentation

Nous employons l'algorithme de segmentation adaptative de carte de profondeur de Parvizi et Wu[64]. Il est adapté à tout type de scène et assez efficace pour envisager une implémentation en temps réel. Cet algorithme repose sur le découpage de la densité de probabilité de profondeur de l'image selon ses extrema locaux et sur un étiquetage en composantes connexes de chacun de ces découpages pour réaliser la segmentation de manière adaptative, c'est-à-dire en ne faisant aucune supposition sur la scène. L'algorithme consiste en quatre étapes :

- filtrage de l'image ;
- estimation de la densité de probabilité de la profondeur ;
- découpage de la densité de probabilité en plusieurs plans ;
- étiquetage en composantes connexes de chacun des plans.

**a Filtrage**

La carte de profondeur  $I_d$  peut être obtenue grâce à plusieurs procédés, mais elle est généralement assez bruitée. Elle est filtrée de manière à retirer le bruit tout en cherchant à conserver les discontinuités de l'image. Nous utilisons un filtre médian, qui consiste à remplacer toutes les valeurs de pixels par la médiane des valeurs de pixels de son entourage. Son entourage est une fenêtre de côté  $2r + 1$ , avec  $r \in \mathbb{N}^*$ , centrée sur le pixel. L'algorithme naïf est peu efficace car le calcul de la médiane demande le tri de  $(2r + 1)^2$  valeurs. En pratique, la bibliothèque logicielle OpenCV utilise un algorithme efficace en temps constant[71].

**b Estimation de densité**

La densité de probabilité de la profondeur doit être estimée à partir de la carte de profondeur. Les méthodes permettant de réaliser l'estimation  $\hat{f}$  d'une densité de probabilité  $f$  à partir d'un échantillon  $(x_1, \dots, x_n)$  sont de deux types : *paramétriques* et *non-paramétriques*. Les méthodes d'estimation paramétriques supposent que  $f$  suit une forme prédéterminée — comme une loi normale, par exemple — et proposent de trouver ses paramètres — dans notre exemple, les paramètres sont la moyenne  $\mu$  et la variance  $\sigma^2$ . Les méthodes d'estimation non-paramétriques sont celles qui ne supposent pas de forme particulière pour  $f$ .

La méthode d'estimation non-paramétrique la plus simple est l'histogramme. L'estimation par histogramme consiste à comptabiliser les occurrences des valeurs de l'échantillon par casiers dont la taille et l'origine est arbitraire. Cette méthode d'estimation possède cependant plusieurs défauts :  $\hat{f}$  n'est pas une fonction continue, et la taille des casiers influe beaucoup sur sa forme.

Une méthode d'estimation non-paramétrique couramment utilisée est l'estimation par noyau. On introduit une fonction  $K$  positive et symétrique, le noyau, telle que  $\int_{-\infty}^{+\infty} K(y)dy = 1$ . Dans cette méthode, l'estimation est donnée par l'accumulation des *contributions locales* de chaque échantillon, qui sont représentées par le noyau. En définissant une *taille de fenêtre*  $h \in \mathbb{R}$ , alors l'estimation de  $f$  par le noyau  $K$  est :

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right)$$

Cette méthode nécessite de choisir une taille de fenêtre et un noyau. La taille de fenêtre a une importance décisive puisque c'est elle qui va déterminer le *lissage* de la densité de probabilité. Un réglage empirique dépendra d'une configuration donnée, mais des méthodes permettent de sélectionner une taille de fenêtre optimale[74]. Le noyau d'Epanechnikov  $K_e(y) = \frac{3}{4}(1 - y^2)$  sur  $[-1; 1]$  est défini comme optimal, mais le choix du noyau n'influe pas beaucoup sur l'estimation et le noyau gaussien  $K_g(y) = \frac{1}{\sqrt{2\pi}}e^{-\frac{y^2}{2}}$  est le plus utilisé car il permet une estimation plus rapide. En effet, l'algorithme naïf de l'estimation par noyau nécessite  $O(nm)$  calculs du noyau pour calculer  $m$  points de  $\hat{f}$ , alors que l'estimation par l'intermédiaire d'une transformée gaussienne rapide n'en nécessite que  $O(n + m)$ . La transformée gaussienne rapide permet de calculer rapidement des fonctions de la forme suivante[28] :

$$S(x) = \sum_{i=1}^n f_i e^{-\frac{(x_i - x)^2}{\sigma^2}}$$

Ici, on peut donc calculer l'estimation par noyau gaussien en posant  $\sigma = h\sqrt{2}$  et  $f_i = \frac{1}{nh\sqrt{2\pi}}$ . Dans la pratique, la bibliothèque logicielle FIGTree[59] implémente la transformée

gaussienne rapide et permet donc également de calculer une estimation par noyau.

### c) Découpage

La densité de probabilité ainsi estimée est ensuite découpée en fonction des minima locaux qui permet de segmenter la scène en autant de plans. Ce découpage est produit la suite  $(m_1, \dots, m_p)$  des points consécutifs où un minimum local est atteint, telle que :

$$\begin{cases} m_1 = \alpha \\ m_i = \arg \min_{m_{i-1} < x < m_{i+1}} \hat{f}(x) & i=2 \dots (p-1) \\ m_p = \beta \end{cases}$$

### d) Étiquetage en composantes connexes

L'image binaire correspondant au plan  $p$  peut être ainsi définie par :

$$I_p(x, y) = \begin{cases} 1 & \text{si } m_p \leq I_d(x, y) < m_{p+1} \\ 0 & \text{sinon} \end{cases}$$

Chaque image binaire  $I_p$  de chaque plan  $p$  extrait est enfin analysée pour séparer les objets qui la composent en un ensemble  $B_p = \{b_{p,1}, \dots, b_{p,n_p}\}$  de groupes de pixels connexes distincts. Les algorithmes qui réalisent cet étiquetage en composantes connexes sont multiples et sont caractérisés par le type de connexité souhaité — *4-connexité* ou *8-connexité*. Deux pixels sont *4-connexes* si l'un est dans le *4-voisinage* de l'autre, le *4-voisinage* d'un pixel étant l'ensemble des quatre pixels situés respectivement un pixel au dessus, en dessous, à gauche et à droite. De la même manière, deux pixels sont *8-connexes* si l'un est dans le *8-voisinage* de l'autre, le *8-voisinage* d'un pixel comprenant son *4-voisinage* complété des pixels situés un pixel au dessus et à gauche, au dessus et à droite, en dessous et à gauche et enfin en dessous et à droite.

À l'issue de cette dernière étape, on a donc segmenté la carte de profondeur :

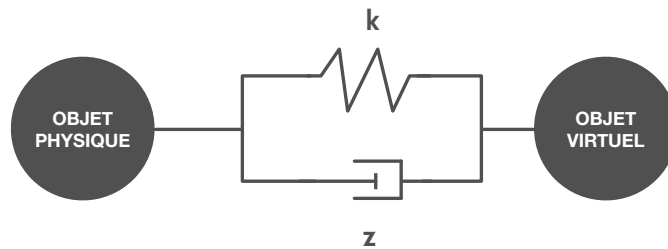
$$C = \bigcup_{i=1}^p \bigcup_{j=1}^{n_p} b_{i,j}$$

## 4. Association et exploitation

Puisque les images  $I_c$  et  $I_d$  sont alignées, chaque objet  $o \in O_V$  détecté par un marqueur dans l'image  $I_c$  peut être associé à un groupe de pixels par une application  $T : O_V \rightarrow C$ . D'autres algorithmes peuvent ensuite être utilisés sur les groupes de pixels concernés pour extraire des informations sur les objets segmentés. Par exemple, l'aire des groupes de pixels peut définir l'aire apparente de la projection des objets sur la caméra, ce qui permet de donner une idée de leurs tailles relatives.

## 5. Conclusion

La méthode que nous proposons permet au système de captation de distinguer les objets qui portent un marqueur ainsi que plusieurs caractéristiques qu'il est possible de tirer de leur apparence. Cependant, plusieurs remarques sont à faire sur cette méthode. Premièrement, nous devons réduire les dimensions des images d'un facteur  $q$  afin d'atteindre des temps de



**FIGURE 4.2** Couplage virtuel

traitement assez courts pour ne pas impacter l'interaction. Cependant, avec des images plus réduites, la précision diminue, et il y a un compromis à faire entre la rapidité d'exécution et la précision des algorithmes. Nous considérons empiriquement que  $q = 16$  convient pour une image de  $640 \times 480$  pixels. Deuxièmement, nous n'implémentons pas de techniques pour régler automatiquement la taille de la fenêtre  $h$  ou la taille du filtre médian. Nous réglons ces paramètres empiriquement en fonction des scènes rencontrées. Troisièmement, les objets trop proches sont fusionnés par l'algorithme, ce qui se produit lorsque quelqu'un prend en main un objet. Cela provient du choix de  $h$  et de la nature de l'algorithme de segmentation, qui traite uniquement la carte de profondeur. D'autres algorithmes de segmentation plus robustes qui utilisent les couples d'images en couleur et de cartes de profondeur existent[11] mais exhibent une complexité plus grande.

### 3 Couplage virtuel

La latence des systèmes de captation basés sur la vision semble handicaper le contrôle musical. Nous verrons dans quelle mesure l'emploi d'un modèle physique particulière peut aider à établir une interaction efficace dans le cadre du contrôle musical, puis nous détaillerons les modalités de simulation d'un tel modèle.

#### 1. Motivations

Nous avons déjà vu qu'un des impératifs absolus est de garder une latence et une gigue assez basses pour que l'interaction reste cohérente, et que la causalité de l'interaction soit perçue[1]. Les exigences sont d'autant plus critiques pour le contrôle musical[85]. Dans les systèmes interactifs, le traitement de la captation occupe une part non négligable du temps d'exécution, surtout lorsque des systèmes de vision artificielle sont employés. Dans ce cadre, la latence et la cadence n'est pas suffisante pour reproduire fidèlement la continuité et la subtilité des mouvements des objets captés. Afin de réduire l'impact de ces problèmes sur le contrôle musical, nous proposons d'introduire une indirection explicite entre les positions des objets captés et celles de leurs représentants virtuels. Un modèle physique d'oscillateur harmonique amorti lie un objet capté et son représentant virtuel, et le mouvement de ce dernier est perçu par l'utilisateur, de manière à rendre explicite ce *couplage virtuel* (fig. 4.2). De cette manière, les mouvements des objets semblent continus et l'utilisateur perçoit la latence du système comme l'effet naturel de la liaison entre les objets couplés. Le couplage virtuel garantit ainsi la causalité de l'interaction pour l'utilisateur qui le perçoit.

Une méthode similaire a déjà été employée pour le rendu haptique, c'est-à-dire l'utilisation d'une interface haptique dotée d'un capteur et d'un actuateur pour simuler des phénomènes réalistes. Le cas de base du rendu haptique est celui du *mur virtuel*, dans lequel l'interface

**FIGURE 4.2**

haptique limite le mouvement à un degré de liberté avec une contrainte unilatérale qui restreint le mouvement jusqu'à un certain point, comme si un mur virtuel empêchait d'aller plus loin. Le problème consiste à réaliser le mur virtuel au contact le plus dur possible de manière à obtenir un réalisme maximal. Si on considère l'*impédance* comme la résistance de l'interface aux manipulations, cela revient à maximiser l'impédance lors du contact avec le mur, et à la minimiser lorsqu'il n'y a pas de contact. Un des phénomènes qui posent problème est que les interfaces peuvent être actives, c'est-à-dire qu'à cause du fait que la captation est échantillonnée et engendre de la latence, l'actuation de l'interface peut générer un travail moteur qui n'est pas naturel. Dans des conditions garantissant la passivité de l'interface[17], une solution proposée au problème du mur virtuel est de disposer d'une impédance élevée et de mettre en œuvre un *couplage virtuel* entre l'interface et sa représentation virtuelle de manière à compenser cette impédance lorsqu'il n'y a pas de contact[18]. Là aussi, le couplage virtuel introduit une indirection qui intègre la latence du système à échantillonnage dans une simulation perçue comme continue.

## 2. Modèle physique

Soient deux masses ponctuelles  $\hat{O}$  et  $O$  (de masse unitaire), représentant respectivement l'objet réel et l'objet virtuel du couplage virtuel, liées par une liaison élastique de raideur  $k$  exerçant une force  $F_{\text{élastique}}$  et une liaison visqueuse d'amortissement  $z$  exerçant une force  $F_{\text{frottement}}$ . Soient  $x_{\hat{O}}$  et  $x_O$  les positions respectives des masses  $\hat{O}$  et  $O$ . On a les forces suivantes :

$$F_{\text{élastique}} = k(x_{\hat{O}} - x_O)$$

$$F_{\text{frottement}} = -z\left(\frac{dx_{\hat{O}}}{dt} - \frac{dx_O}{dt}\right)$$

La position de  $\hat{O}$  est totalement déterminée par l'objet réel. On considère que sa position n'est pas soumise aux forces. Cependant, la position de  $O$  l'est, et la résultante  $\sum F$  des forces appliquées à  $O$  est :

$$\sum F = k(x_{\hat{O}} - x_O) - z\left(\frac{dx_{\hat{O}}}{dt} - \frac{dx_O}{dt}\right)$$

La deuxième loi de Newton permet de calculer l'accélération de  $O$  :

$$\sum F = \frac{d^2x_O}{dt^2}$$

## 3. Méthode

On réalise une simulation discrète en temps du système décrit ci-dessus avec un pas de temps de 1. On considère  $\hat{x}_n$  la position de  $\hat{O}$  à l'instant  $n$ , et de même,  $x_n$  est la position de  $O$  à l'instant  $n$ . De la même manière, on aura  $\hat{v}_n$  et  $\hat{a}_n$  respectivement la vitesse et l'accélération à l'instant  $n$ . À l'instant  $n$ ,  $\sum F$  s'écrit alors :

$$\sum F = k(\hat{x}_n - x_n) - z(\hat{v}_n - v_n)$$

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

La deuxième loi de Newton donne, pour  $O$  et pour l'instant  $n + 1$  :

$$\begin{aligned}\sum F &= a_{n+1} \\ \sum F &= v_{n+1} - v_n \\ \sum F &= x_{n+1} - 2x_n - x_{n-1}\end{aligned}$$

Ce qui nous permet de calculer la prochaine position de  $O$  avec ses anciennes positions et l'expression de  $\sum F$  à l'instant  $n$  :

$$x_{n+1} = x_n + v_n + \sum F$$

La simulation consiste donc à répéter, à haute fréquence, et avec des positions et des forces qui sont des vecteurs tridimensionnels, le cycle d'opérations suivant :

- calcul des nouvelles positions ;
- calcul des nouvelles forces.

## 4. Conclusion

Le choix de la fréquence de la simulation et des paramètres  $k$  et  $z$  détermine le comportement du couplage virtuel et peut être guidé par l'étude des différents régimes de l'oscillateur harmonique amorti. Un  $z$  trop important augmente artificiellement la latence et rend impossible l'exécution de mouvements à haute fréquence. Lorsque  $z$  est trop faible et que  $k$  est trop important, le système suit les discontinuités de la captation et perd son intérêt. Lorsque  $k$  et  $z$  sont trop faibles, le système devient instable et les objets virtuels oscillent continuellement autour de leurs contreparties physiques. Il faut donc régler les paramètres en fonction des performances de la captation et de l'effet recherché. Dans tous les cas, l'utilisation d'un modèle physique donne au couplage virtuel un aspect déterministe et réaliste auquel il est possible de s'habituer. Cela laisse envisager qu'un utilisateur expérimenté peut maîtriser cet aspect temporel pour un contrôle musical plus efficace. Nous doutons cependant qu'il est possible de s'adapter au couplage virtuel dans le cas des gestes d'excitation percussive.

# 4 Techniques d'interaction opportunistes

Dans cette section, nous présentons quelques techniques d'interaction pertinentes pour la construction musicale opportuniste. Après un exposé de nos motivations, nous décrirons des techniques existantes, puis détaillerons les méthodes que nous avons conçues à partir de ces techniques.

## 1. Motivations

On a établi que les interfaces permettant la reconfiguration dynamique pouvaient reposer sur les propriétés de liaison dynamique et de composition dynamique.

Pour la liaison dynamique, deux cas de figure sont envisagés : soit cette liaison est implicite et résulte de choix déterminés par des heuristiques si possibles connues par les utilisateurs, soit elle est explicite et l'utilisateur doit pouvoir communiquer ses intentions à cet égard. Le cas de la liaison dynamique implicite reviendrait par exemple à dire que si un objet rouge est utilisé, alors il représentera une note, tandis que les objets verts représenteront le mécanisme de copie. La liaison dynamique implicite semble aller à l'encontre de l'opportunisme :

les mécanismes dépendent de paramètres que l'utilisateur peut ne pas contrôler — il suffit d'imaginer utiliser l'interface décrite par l'exemple précédent dans une jungle. Le cas de la liaison dynamique explicite nécessite des techniques d'interaction qui permettent à l'utilisateur de choisir le couplage à réaliser.

Pour la composition dynamique, le même dilemme se pose : l'introduction des nouveaux objets se fait de manière implicite ou explicite. Le cas implicite n'est souhaitable que dans le cas où une instrumentation des objets physique par l'utilisateur est nécessaire, et dans ce cas c'est le fait de détecter ces objets instrumentés qui marque leur inclusion dans le système. Le cas explicite nécessite des techniques d'interaction qui permettent à l'utilisateur de signifier l'entrée ou la sortie d'un objet du contexte interactif.

La reconfiguration dynamique opportuniste nécessite donc de déterminer deux types de techniques d'interaction qui correspondent aux deux propriétés que nous avons identifiées.

## 2. Techniques pour la liaison dynamique opportuniste

Dans le cadre de la liaison dynamique opportuniste, une représentation, ou un objet virtuel à associer doit pouvoir être sélectionné parmi un ensemble de choix qui peut varier. Les techniques existantes ne sont pas forcément adaptées à ce contexte précis mais ont tous le même fonctionnement inspiré du menu.

Par exemple, le système *Tiles* propose d'utiliser un *catalogue* de liaisons possibles, sous la forme d'un vrai livre où chaque page représente un objet virtuel qu'il est possible de lier à un objet réel[72]. Pour concrétiser la liaison dynamique d'un des objets virtuels avec un objet réel, l'utilisateur place ce dernier près de la page et attend une seconde. De manière similaire, il réalise la même opération pour enlever la liaison. Ce catalogue est similaire à un menu de restaurant, une métaphore qui a également inspiré les menus des interfaces graphiques. De nombreux types de menus différents ont été réalisés pour les interfaces graphiques, la réalité virtuelle et la réalité augmentée[13]. Une taxonomie récente, établie dans [23], permet de classer les différents types de menus en trois dimensions selon plusieurs critères : la nature des éléments (leur nombre et leur hiérarchie), l'apparence et la structure du menu (structure géométrique, organisation, type de données affichées, taille et espacement des éléments de menu), la position du menu, ses modalités d'appel, la configuration de l'interaction (périphérique d'interaction utilisé, domaine d'application, retour, etc.), l'utilisabilité et la combinabilité (la possibilité d'être utilisé avec d'autres types de menus).

Si cette taxonomie doit permettre de guider la conception de menus dans les applications, nous souhaitons nous concentrer uniquement sur certaines caractéristiques plus spécifiques aux besoins de la construction musicale. Des techniques d'interaction pour la liaison dynamique dans ce cadre ont été mises au point pour l'interface tangible *Audiopad*[68]. Une des techniques concernait la sélection et la navigation bimanuelle dans un menu hiérarchique. L'invocation du menu est réalisée en approchant un objet spécifique en forme d'étoile (le *selector puck*) d'un objet d'intérêt (un *puck*) auquel on veut associer un échantillon sonore. Le menu s'affiche de manière circulaire autour du *selector puck* mais reste fixe, et le déplacement de ce *puck* sur un des éléments textuels du menu provoque l'affichage des éléments de niveau inférieur. Cette technique d'interaction est efficace, notamment grâce à l'utilisation des deux mains et de la forme circulaire du menu. Les menus circulaires, ou *radiaux*, sont souvent utilisés en raison du fait qu'ils sont affichés au même endroit qu'ils sont invoqués[86].

### 3. Techniques pour la composition dynamique opportuniste

Les techniques d'interaction pour la composition dynamique opportuniste doivent permettre à l'utilisateur d'exprimer son souhait d'impliquer un objet dans l'interaction. On peut supposer que cette étape précède l'affichage d'un menu qui permet alors de préciser le rôle de l'objet dans le système. Deux méthodes peuvent être envisagées.

L'une d'elles s'appuie sur un objet spécial, comme le *selector puck* de l'*Audiopad* ou le *catalogue* des *Tiles*, dont le rôle est de sélectionner un objet et de confirmer sa sélection. Dans ceux deux exemples, la sélection est faite en rapprochant l'objet spécial de l'objet à impliquer et la confirmation est faite en restant dans cette configuration spatiale pendant un temps limité. L'autre méthode s'appuie sur la réalisation d'un geste précis avec l'objet à impliquer. Par exemple, un *shake menu* s'affiche lorsqu'un objet reconnu est secoué[86]. Le menu est radial, et la sélection d'un élément se fait en positionnant l'objet à proximité de l'élément de menu sélectionné, puis en secouant l'objet encore une fois. Les différents positionnements possibles du menu (par rapport à l'objet, par rapport à l'affichage ou par rapport à l'environnement physique) sont comparés.

### 4. Méthode

Nous avons conçu deux méthodes inspirées des techniques d'interaction évoquées plus haut : la méthode *objet alpha*, et la méthode *shake menu* (fig. 4.3). Toutes les deux font appel à un menu radial et permettent d'afficher un nombre arbitraire d'éléments.

**FIGURE 4.3**

#### a) Objet alpha

Dans cette approche, un objet spécifique est désigné comme l'*objet alpha* avant l'exécution du système. Cet objet est un objet spécial qui permet d'impliquer les autres objets détectés et de réaliser des opérations de contrôle d'application.

Lorsqu'un utilisateur souhaite donner un rôle à l'objet de son choix dans le système, il s'assure que l'objet en question est visible un instant, puis place l'objet alpha devant. En attendant un court instant dans cette configuration spatiale, un disque coloré centré sur l'objet alpha commence à apparaître. Une indication visuelle demande de déplacer l'objet alpha de manière à rendre à nouveau visible l'objet concerné. Une fois ceci fait, un menu radial centré sur l'objet alpha remplace le disque. La rotation de l'objet alpha détermine l'élément du menu qui est actuellement sélectionné, et la sélection courante dans le disque est mise en évidence. La confirmation de cette sélection se fait en déplaçant l'objet alpha à nouveau sur l'objet concerné.

#### b) Shake menu

Dans cette approche, chaque objet a le pouvoir de faire apparaître un menu tant qu'il n'est pas impliqué d'une autre façon dans le système. Lorsqu'un objet est secoué suffisamment et assez rapidement, un disque coloré centré sur l'objet commence à apparaître en restant attaché à l'objet. Au bout d'un certain temps, le disque est remplacé par un menu radial composé d'au plus huit éléments. Ce menu n'est plus attaché à l'objet mais à l'affichage. L'objet sert alors de pointeur pour sélectionner soit un élément du menu radial, soit un élément permettant de naviguer entre les différentes pages du menu, dans le cas où il présente plus de huit entrées. Une fois qu'un élément est sélectionné, la sélection ne change pas à moins qu'un autre élément soit sélectionné ou que l'objet soit remis au centre du menu radial. Dans ce dernier cas, la sélection est effacée. Une autre secousse permet de confirmer la sélection.



Pour un objet dont la vitesse instantanée à l'instant  $n$  est  $\vec{v}_n$ , on détecte les secousses d'un objet par un seuil sur le nombre de changements de signe du produit scalaire  $\vec{v}_n - \vec{v}_{n-1}$  dans un intervalle de temps fixé. Une temporisation après la secousse associée à un retour visuel progressif (le disque commence à apparaître) permet à l'utilisateur de se rendre compte que son geste a été pris en compte.

## 5. Conclusion

Nous avons déterminé que la reconfiguration dynamique demandait des techniques d'interaction explicites. Sous l'angle de la liaison dynamique, nous avons établi que ces techniques devaient exploiter la métaphore du menu. Sous l'angle de la composition dynamique, nous avons distingué les méthodes s'appuyant sur un objet spécial, et celles où chaque objet se manifeste à l'aide d'un geste. Nous avons donc conçu deux méthodes illustrant ces différences. L'approche *objet alpha* est globalement satisfaisante et le choix d'un bon objet alpha permet une sélection rapide, mais la méthode de sélection du menu par rotation de l'objet alpha est fastidieuse et n'est pas appropriée à toutes les situations. L'approche *shake menus* est également satisfaisante et autorise des manipulations simultanées par plusieurs utilisateurs, mais elle est difficile à utiliser si l'objet n'a pas tous ses degrés de liberté. Une étude approfondie et peut-être un mélange des deux approches sont souhaitables pour étudier cet aspect de l'interaction opportuniste.

Dans ce chapitre, nous détaillons les concepts que nous avons définis durant ce stage, la réalisation qui les implémente, puis nous présentons des scénarios qui nous permettront d'émettre des observations quant à la réalisation et son impact sur notre objet d'étude. Enfin, nous exposons quelques perspectives pour de futurs travaux sur ce sujet.

## 1 Un système de construction opportuniste

Nous avons décidé de réaliser ce qui s'apparente à une première étape vers un système configurable de construction et de performance opportuniste. Un tel système doit permettre à un utilisateur de construire, gérer et manipuler des processus musicaux par l'intermédiaire d'objets de son environnement immédiat, dans le cadre d'une performance musicale. Nous décrivons notre système en définissant tout d'abord des notions qui ensemble constituent un cadre conceptuel dans lequel la réalisation ensuite décrite s'inscrit.

### 1. Environnement, processus musicaux et comportements

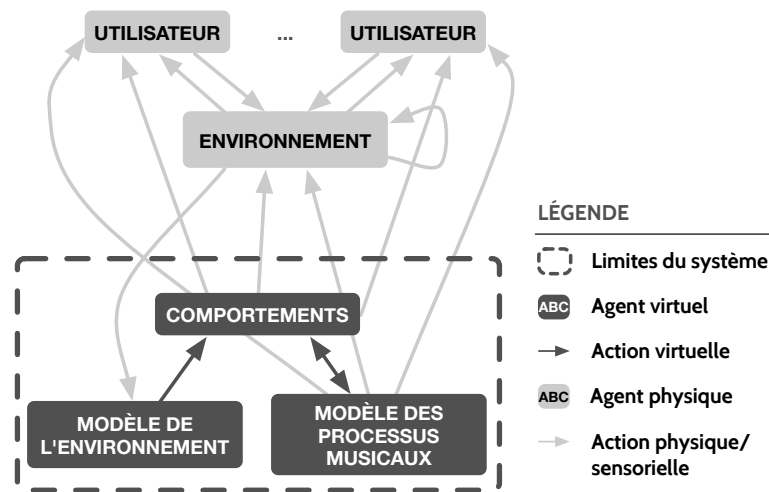
Pour le moment, ce que nous avons décrit des interfaces de construction musicale opportunistes ne donne aucune indication sur la manière de construire un tel système. Nous introduisons donc ici quelques concepts qui précisent les modalités de l'interaction avec le système, et qui serviront de base pour la réalisation de notre système.

Le système reçoit des informations de l'environnement immédiat et les interprète afin de construire un **modèle de l'environnement**. Ce modèle est constitué d'un ensemble d'**objets**. Chaque objet représente un artefact physique qui a été identifié et dont certaines propriétés ont été perçues grâce aux informations provenant de l'environnement immédiat. Un objet est donc muni de propriétés, comme son identité, sa position dans l'espace par rapport à un repère donné, ses dimensions, sa forme, sa couleur... Le modèle de l'environnement est continuellement mis à jour pour représenter ce que le système *sent* ou *voit* du monde extérieur à chaque instant.

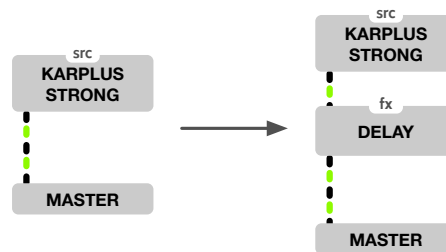
D'autre part, le système doit gérer un ensemble de **processus musicaux** de natures diverses, et les relations qu'ils peuvent avoir entre eux. Ce **modèle des processus musicaux** qu'il entretient à chaque instant également.

L'enjeu est alors de pouvoir permettre à des utilisateurs de contrôler les liaisons entre des objets, d'une part, et des processus musicaux, d'autre part. Il existe énormément de manières de concevoir ces liaisons, et la nature de ces liaisons conditionne le pouvoir expressif musical du système. Il est également souhaitable de pouvoir lier des objets à des fonctions qui n'agissent pas directement sur les processus musicaux, mais sur la gestion des liaisons ou d'autres aspects du système, par exemple.

Nous rendons toutes ces liaisons possible en introduisant la notion de **comportement**. Un comportement est ce qui définit les réponses du système aux variations des propriétés



**FIGURE 5.1** Interactions dans et autour du système



**FIGURE 5.2** Insertion implicite d'un effet après une source sonore

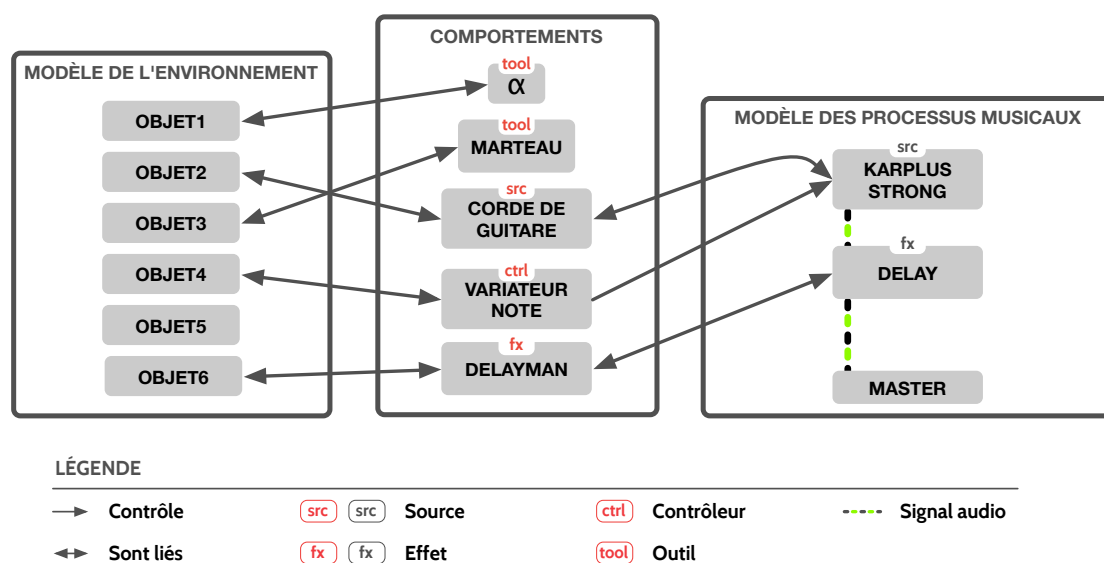
d'un objet. Un comportement est *attaché* à un seul objet, et un objet peut avoir un seul comportement ou aucun. Cette définition est volontairement vague, car elle inclut de nombreuses possibilités. Le comportement peut définir la manière dont l'objet est augmenté visuellement, la manière dont il agit sur des processus musicaux existants (en fonction, par exemple, des mouvements de l'objet auquel il est attaché), ou sur des nouveaux processus musicaux dont il a la responsabilité (l'objet *représente* alors un ou plusieurs processus musicaux, du point de vue de l'utilisateur), ou bien encore sur d'autres comportements (des comportements peuvent par exemple servir à en changer d'autres, ou à en attacher à des objets). C'est une entité qui symbolise la part de représentation ou de traitement virtuel attachée à un objet et qui définit une manière de réagir dans le temps face à des manipulations.

Les interactions entre les trois parties du système — un modèle de l'environnement, un modèle des processus musicaux, et un ensemble de comportements — et l'environnement physique, qui sont schématisées sur la figure 5.1, résument le fonctionnement d'une interface de construction musicale opportuniste.

**FIGURE 5.1**

## 2. Gestion transparente des processus musicaux

Il existe plusieurs paradigmes pour les structures de production sonore numérique. Les stations de travail audio numériques utilisent couramment le paradigme de la table de mixage, avec ses tranches, ses bus, ses départs et retours. Les environnements de synthèse modulaire utilisent le paradigme du traitement du signal, et proposent de câbler le trajet du signal audio entre différentes unités de génération ou de traitement jusqu'à des sorties. C'est précisément ce paradigme que nous avons employé pour représenter et manipuler les processus



**FIGURE 5.3** Liens entre objets, comportements et processus musicaux

sonores, que nous définirons ici comme des unités de génération ou de traitement du signal audio. Nous nous sommes limités à une sortie audio, et au plus une entrée et une sortie par processus musical, avec une sortie principale monophonique (*master*). Nous distinguons deux types de processus musicaux qu'il est possible de manipuler : les *sources*, générateurs de son, qui n'ont pas d'entrée et une sortie, et les *effets*, qui transforment le signal, et ont donc une entrée et une sortie. Le détail du fonctionnement interne de la synthèse se trouve à la page 40.

**FIGURE 5.2**

Dans notre interface, la gestion de connexions entre les processus musicaux est implicite. Lorsqu'un comportement ayant la responsabilité d'un nouveau processus musical est créé, la structure des processus musicaux est automatiquement mise à jour : le nouveau processus musical est relié par défaut au *master*. Un effet est placé par rapport à une source ou un autre effet, et sa sortie est reliée au *master*, remplaçant la connexion qui existait précédemment (figure 5.2). Le fait que la gestion de ces connexions soit implicite simplifie l'accès au système et son utilisation, mais demande une réflexion sur les techniques d'interaction qui permettent une gestion avancée. Par exemple, comment spécifier qu'un effet de réverbération se trouvera, dans le trajet du signal, entre un oscillateur sinusoïdal et un filtre passe-bas ? Faute de temps, nous avons choisi de ne pas traiter ces questions. Cependant, certains travaux existants peuvent fournir des pistes de réflexion. Sur la *reactTable*, par exemple, c'est la proximité entre les éléments qui définit les connexions[51].

### 3. Conception des comportements

Les comportements sont de quatre sortes : les *sources* et les *effets*, qui représentent respectivement une source ou un effet (les processus musicaux), les *contrôleurs*, qu'il faut attacher à autre un comportement pour pouvoir contrôler un ensemble de paramètres, et les *outils*, qui regroupent le reste des possibilités. Un *outil* peut être utilisé pour agir sur d'autres comportements de manière ponctuelle, sans y être attaché, comme un marteau sur des clous.

#### a Mise en situation

**FIGURE 5.3**

La figure 5.3 présente les liens entre les objets, les comportements et les processus musi-

caux à travers un exemple de construction. Dans ce scénario, l'utilisateur voit six objets dans son champ de vision, dont cinq sont associés à des comportements :

- un *objet*  $\alpha$ , qui est un outil implémentant la technique d'interaction éponyme vue au chapitre 4, et qui permet d'attacher des comportements à des objets ;
- un *marteau*, qui est un outil qui envoie une impulsion à tous les objets attachés à une source avec lesquels il rentre en collision ;
- une *corde de guitare*, qui est responsable d'une source générant des sons de cordes pincées en réponse à des impulsions, grâce à l'algorithme de Karplus-Strong[52] ;
- un *variateur note*, qui est un contrôleur qui assigne la position de l'objet attaché selon un axe donné à la note émise par la source ou l'effet auquel il est attaché ;
- un *delayman*, qui est responsable d'un effet sonore de type écho, qui traite le signal audio sortant du générateur de sons de cordes pincées.

Avec ces éléments, l'utilisateur a construit un petit instrument, similaire à une guitare à une seule corde, dont il peut déjà jouer. Son souhait de contrôler d'autres aspects de la synthèse peut l'amener à trouver d'autres objets pour incarner des contrôleurs. Il peut également décider d'enrichir son environnement sonore en rajoutant d'autres sources et effets.

### **b** Importance de la conception des comportements

On voit que, dans ce système, ce sont donc les comportements qui vont définir la plus grande partie de l'interaction : la manière dont l'utilisateur va construire la structure de production sonore, mais également la manière dont les objets peuvent être utilisés pour le contrôle des processus sonores. Ils est donc de la plus grande importance de pouvoir proposer des comportements variés et veiller, dans le processus de conception, à faciliter la construction ou la performance musicale.

Nous implémentons les techniques d'interaction pour la construction proposées plus haut sous forme de comportements. De nombreux travaux peuvent guider la conception des comportements destinés à la performance musicale, ces comportements déterminant largement la qualité du contrôle[42]. Premièrement, des modèles de mapping[2, 36] permettent d'identifier les liaisons pertinentes entre des paramètres de contrôle, que le musicien manipule, et des paramètres de synthèse. Deuxièmement, des modèles d'interaction, comme celui de l'interaction instrumentale[4], permettent de gérer la complexité de l'interaction qui peut être mise en œuvre dans le cadre du contrôle opportuniste de la musique. Utilisés de manière complémentaire, ces modèles permettent de concevoir un contrôle musical plus efficace[22].

Dans le cadre de ce stage, nous ne nous intéressons pas à la conception des comportements, mais nous insistons sur le fait que c'est un point crucial et que la problématique du mapping n'a pas été développée dans les précédents travaux sur le contrôle opportuniste de la musique[54].

## **2** Implémentation

Notre réalisation est implémentée en deux parties : un programme principal, écrit en C++, et un programme de synthèse sonore, réalisé avec Puredata.

Le programme principal est constitué de plusieurs parties :

- gestion du modèle des processus sonores ;
- gestion de bas niveau de la captation ;
- interprétation de la captation (*sondes environnementales*) ;
- gestion du modèle de l'environnement ;
- gestion des comportements ;

- couplage virtuel ;
- affichage ;
- exécution ;
- implémentation des techniques d'interaction.

Nous présentons les détails du programme de synthèse sonore, puis abordons ensuite la réalisation technique de chacune des parties du programme principal. Nous terminerons par détailler l'implémentation des techniques d'interaction pour la construction opportuniste.

## 1. Programme de synthèse sonore

Le programme de synthèse sonore est construit grâce à *Puredata*[73], un environnement de programmation visuelle modulaire dédié au développement d'applications multimédia et créé par Miller Puckette.

Dans *Puredata*, un *patch* est la spécification d'un programme de traitement d'informations numériques et de flux de signaux, qui est constituée de blocs et de connexions qui représentent respectivement les transformations et le trajet des données. Le programme ainsi décrit peut être exécuté et modifié en temps réel. La création de *patches* de haut niveau se fait en combinant des programmes réutilisables de plus bas niveau, qui peuvent être des *abstractions* (d'autres patches indépendants) ou des *externals* (programmes compilés et écrits dans d'autres langages).

Plusieurs parties concourent à la synthèse sonore. Un *patch de gestion* est chargé de gérer les événements entrants et de traiter les composantes de la synthèse et leurs connexions en conséquence. Plusieurs *patches secondaires* constituent les différents processus sonores qu'il est possible d'utiliser, et qui suivent une convention bien particulière. Enfin, des *règles* de gestion des paramètres des processus sonores et de leurs connexions sont appliquées depuis le programme principal par l'intermédiaire d'un protocole de communication.

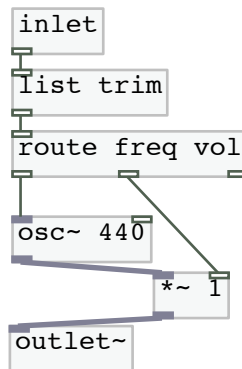
### a Gestion des événements et des connexions

Le patch de gestion reçoit les événements de l'extérieur et les interprète d'une part, et gère dynamiquement les connexions entre les différentes parties de la structure de production sonore d'autre part.

La gestion des connexions et des éléments qui font partie de la synthèse est faite grâce à l'*external dyn~*[37]. Cet external permet de gérer un sous-patch et de modifier dynamiquement ses composants, ses connexions, ou d'envoyer des messages à ses objets. C'est lui qui permet de réaliser techniquement les règles de connexion des processus sonores que nous avons évoqué plus haut.

La réception des événements est assurée par l'intermédiaire du protocole de communication OSC[87] (*Open Sound Control*) à travers le protocole de transport UDP. Les unités de base de la communication dans OSC sont des *messages*. Un message est composé d'une *adresse*, qui est une chaîne de caractères formatée comme un chemin UNIX, et d'une suite de *paramètres* pouvant prendre des types différents (nombre entier, nombre à virgule, chaîne de caractères, etc.). Chaque message OSC reçu est ensuite interprété. Trois adresses sont reconnues par le patch :

- */semicolon*, qui interprète les paramètres qui suivent comme une commande à destination de *Puredata*. Ce dernier peut être ainsi commandé à distance.
- */dyn*, qui interprète les paramètres qui suivent comme une commande à destination de *dyn~*.
- */dyndelay*, qui interprète le premier paramètre comme un délai en millisecondes, et les paramètres qui suivent comme une commande à destination de *dyn~* qui sera



**FIGURE 5.4** Un exemple de patch secondaire.

exécutée après ce délai.

Ainsi, le programme principal contient une partie de la logique destinée au programme `Purdata` et à son `external dyn~`. Si cette pratique n'est pas recommandée pour développer un logiciel en production, elle convient parfaitement pour un prototype.

### **b** Convention pour les patches secondaires

Chaque patch secondaire est construit de manière identique pour garantir le bon fonctionnement de toutes les combinaisons possibles de processus musicaux.

Les règles évoquées plus haut concernant la connectivité des processus sonores *sources* ou *effets* sont respectées. En plus de cela, chaque patch secondaire possède une entrée de contrôle pour régler les paramètres dont le nom suivi de la nouvelle valeur sont envoyés par message à ce patch secondaire. Nous avons essayé de garder les noms de paramètres quand il était possible de le faire, de manière à permettre la création de comportements pouvant fonctionner sur un grand nombre de processus sonores différents. La figure 5.4 présente l'oscillateur sinusoïdal, un des patches secondaires les plus simples. Voici quelques paramètres avec leur description et un exemple d'utilisation pour chacun :

**FIGURE 5.4**

- `bang` : une impulsion (utilisé pour le déclenchement comme pour l'excitation percussive).
- `play` : un ordre de lecture ou d'arrêt. `play 0` donne l'ordre d'arrêter le son.
- `wet` : le taux de présence d'un effet, entre 0 et 1. `wet 0.5` règle la sortie comme le mélange de 50% du signal d'entrée et de 50% du signal modifié.
- `vol` : le volume d'un processus sonore, entre 0 et 1. `vol 1` règle le volume à son niveau maximal ;
- `freq` : la fréquence, en Hertz. `freq 440` règle la fréquence sur le La du diapason.
- `note` : la note, en *midi pitch*, c'est-à-dire un nombre de demi-tons. `note 69` règle la note sur le La du diapason.

## 2. Programme principal

### **a** Gestion du modèle des processus sonores

Le programme conserve un modèle des processus sonores qui ont été manipulés pendant l'exécution. Chaque patch secondaire du programme de synthèse sonore est modélisé dans le programme principal. Les modifications qu'apporte le programme au modèle des processus

sonores est répercuté dans le patch principal par l'intermédiaire de messages OSC décrits plus haut.

Les différents paramètres possibles sont également modélisés et peuvent être gérés de manière dynamique par des listes de paramètres mutables. Cette infrastructure permet d'envisager des transformations automatiques entre paramètres contrôlées par des comportements de haut niveau.

### **b** Gestion du modèle de l'environnement

Le programme conserve un modèle de l'environnement physique immédiat au fur et à mesure que le programme s'exécute et que les capteurs y apportent de nouvelles informations de manière active (toutes les informations disponibles sont retransmises) ou passive (seules les nouvelles informations sont retransmises). Il gère automatiquement le référencement d'un nouvel objet inconnu et son suivi.

Chaque objet est une structure de données comprenant les caractéristiques suivantes : identifiant unique, position en trois dimensions, rotation sur un axe, absence (pas de mise à jour au-delà d'un temps prédéfini), zombification (absence au-delà d'un temps défini), boîte englobante et comportement attaché.

### **c** Gestion de bas niveau de la captation

Nous permettons la captation depuis une de ces quatre sources :

- une webcam reconnue par la bibliothèque logicielle de capture de Mac OS X, qui permet d'exploiter une image en couleurs ;
- une Microsoft Kinect, par l'intermédiaire de la bibliothèque logicielle *Freenect*, qui permet d'exploiter des couples constitués d'une image en couleurs et d'une carte de profondeur non alignées ;
- une Microsoft Kinect, par l'intermédiaire de la bibliothèque logicielle *OpenNI*, qui permet d'exploiter des couples constitués d'une image en couleurs et d'une carte de profondeur alignées ;
- le protocole de communication TUIO[50], qui permet de transmettre des informations sur des objets provenant d'une interface tangible sur table, comme la *reacTable*, sur le réseau.

Nous avons voulu que la réalisation puisse supporter une caméra couleur, un module de type Kinect ou un système de captation compatible avec TUIO sur une autre machine, de manière à pouvoir nous appuyer, durant le développement, sur une captation précise ou à haute fréquence avec une caméra de qualité, sur l'algorithme de segmentation pour le suivi de marqueurs avec le module Kinect ou sur des machines où aucune caméra n'est disponible, grâce à un simulateur pour TUIO.

### **d** Interprétation de la captation

Les modules en charge de l'interprétation sont appelés *sondes environnementales*. Nous en avons développé quatre :

- la sonde *artoolkitplus*, qui utilise la bibliothèque logicielle ARTToolkitPlus, pour mettre à jour les caractéristiques d'objets repérés par des marqueurs matriciels dans une image en couleurs ;
- la sonde *reactivision*, qui utilise la bibliothèque logicielle reacTIVision, pour mettre à jour les caractéristiques d'objets repérés par des marqueurs topologiques dans une image en couleurs ;

- la sonde *depth\_segmentor*, qui implémente l'algorithme de segmentation pour le suivi de marqueurs détaillé au chapitre 4 pour mettre à jour les caractéristiques des objets qui ont déjà été détectés ;
- la sonde *tuio*, qui interprète des messages TUIO pour mettre à jour les caractéristiques de certains objets.

Les quatre sondes fonctionnent de manière similaire. Elles interprètent des éléments de bas niveau de la captation et mettent à jour le modèle de l'environnement.

### e) Gestion des comportements

Un comportement est obligatoirement attaché à un objet et ne peut pas en changer dans le cours de sa vie. Chaque comportement peut dessiner sur l'affichage et a la possibilité de réaliser n'importe quelle action.

Les comportements appartiennent à quatre catégories : sources, effets, contrôleurs et outils. Les liens entre les comportements sources ou effets, les processus sonores correspondants et leurs paramètres sont gérés automatiquement de manière à pouvoir accepter les associations valides qui peuvent être proposées par l'utilisateur.

### f) Couplage virtuel

Le couplage virtuel vient remplacer les données de position des objets dans le modèle de l'environnement. La simulation est assurée par un processus léger qui met à jour le couplage virtuel (calcul des positions, puis calcul des forces) pour chaque objet du modèle de l'environnement à une fréquence de 100 Hz. Les paramètres de notre simulation sont  $k = 0.08$  et  $z = 0.05$ .

### g) Affichage

Le programme affiche une image couleur provenant d'une caméra, s'il en existe une, et y superpose les affichages de chaque comportement. L'affichage est réalisé grâce aux bibliothèques logicielles Cairo et OpenGL.

### h) Exécution

Nous utilisons la bibliothèque logicielle Cinder pour garantir une exécution d'une fréquence quasi-constante de 30 tours par secondes. À chaque tour, le programme :

- demande à mettre à jour ce qui provient de la captation de bas niveau (images, etc.) ;
- exécute chaque sonde pour traiter ces nouvelles informations de manière à mettre à jour le modèle de l'environnement ;
- pour chaque objet du modèle de l'environnement attaché à un comportement, exécute ce dernier ;
- affiche l'image et y superpose, pour chaque objet du modèle de l'environnement attaché à un comportement, l'image dessinée par ce dernier.

### i) Techniques d'interaction pour la construction opportuniste

Les deux propositions de techniques d'interaction pour la construction opportuniste présentées au chapitre 4 sont implémentées sous forme de comportements (fig. 5.5).

Ces comportements ont la responsabilité d'afficher des menus non hiérarchiques. Un élément de menu est déterminé par un titre et une action, et un menu est déterminé par un

titre et une suite d'éléments de menu. Le menu qui est affiché dépend de la situation. Lorsqu'un objet non attaché à un comportement est concerné, un menu de liaison dynamique est utilisé. Il propose une liste de comportements qu'il est possible d'attacher à l'objet. Si un objet attaché est concerné, un menu propose de détacher le comportement de l'objet.

**Objet alpha** Le premier objet qui est détecté au commencement du programme est attaché à un comportement de type *objet alpha*.

Ce comportement implémente l'automate fini de la figure 4.3. Lorsqu'un menu à  $n$  éléments est affiché, l'indice  $i$  de l'élément de menu sélectionné est déterminé par la rotation  $r \in [0; 2\pi]$  de l'objet alpha (avec  $\lfloor \cdot \rfloor$  la fonction partie entière) :

$$i = \left\lfloor \frac{r}{2\pi} n \right\rfloor$$

**Shake menu** Le comportement *shake menu* implémente l'automate fini de la figure 4.3 et est attaché à tous les nouveaux objets détectés. Afin que le système reste cohérent, nous avons fait en sorte qu'un comportement puisse être provisoire. Un comportement provisoire, comme le comportement *shake menu*, a pour but d'être remplacé par un comportement permanent.

**Assignment pour les effets et contrôleurs** Les comportements de type *contrôleur* doivent être associés à un comportement de type *source* ou *effet*. Cette association est gérée par tous les comportements de type *contrôleur* d'une façon similaire à l'objet alpha : attacher un contrôleur à un comportement se fait en présentant le contrôleur devant le comportement choisi pendant un certain temps, puis en déplaçant le contrôleur de manière à rendre à nouveau visible le comportement choisi.

### 3 Scénarios et observations

Cette section présente les exemples d'utilisation que nous avons réalisés. Nous décrivons les processus sonores et les comportements que nous avons développés jusqu'à là, puis nous détaillons trois scénarios et montrons comment ils illustrent les concepts développés dans ce document.

#### 1. Description des processus sonores et comportements implémentés

Nous décrivons ici les sources sonores, les effets sonores et les comportements que nous avons développés. Nous n'avons pas suivi de démarche particulière pour la conception des processus sonores, mais nous encourageons le développement d'ensembles cohérents de processus sonores pour notre système, ce qui peut faire l'objet d'un développement ultérieur.

##### a Sources sonores

**oscillateur** Un processus de synthèse par oscillateur sinusoïdal contrôlable avec les paramètres *freq* et *vol*.

menu attaché à l'objet



menu attaché à l'affichage



**FIGURE 5.5**

Les méthodes *objet alpha* (en haut), et *shake menus* (en bas), dans la phase de liaison dynamique

**looplayer** Un lecteur d'échantillon sonore en boucle contrôlable avec les paramètres `bang` et `play`.

**karplus\_strong** Un processus de synthèse de sons de cordes pincées par l'algorithme de Karplus-Strong contrôlable avec les paramètres `bang`, `note` et `damp`.

**line\_in** Un processus qui reproduit le signal entrant par par la première entrée audio disponible du système (en général, le microphone de la carte audionumérique).

### **b** Effets sonores

**reverb** Un effet utilisant le moteur d'effets de réverbération *Freeverb* contrôlable avec le paramètre `wet`.

**simple\_delay** Un effet d'écho simple avec rétroaction contrôlable avec les paramètres `wet`, `delay` et `feedback`.

### **c** Comportements

**oscillathor** Un comportement *source* représentant *oscillateur* qui associe la fréquence de ce dernier à une valeur proportionnelle à la position de l'objet attaché sur l'axe horizontal.

**corde guitare** Un comportement *source* représentant *karplus\_strong*.

**micro** Un comportement *source* représentant *line\_in*.

**variateur (note|frequence|volume)** Une suite de comportements *contrôleur* qui permettent de contrôler un paramètre (`note`, `freq` ou `vol`) en l'associant à une valeur proportionnelle à la position de l'objet attaché sur l'axe vertical.

**marteau** Un comportement *outil* qui déclenche un paramètre `bang` sur les comportements compatibles se trouvant à proximité immédiate.

**tete de lecture** Un comportement *outil* qui déclenche un paramètre `bang` sur tous les comportements situés sur une droite dont la direction est déterminée par son angle de rotation.

## 2. Scénarios

La nature de la construction musicale opportuniste permet de reproduire les scénarios suivants, en les construisant par le système avec des objets de l'environnement immédiat.

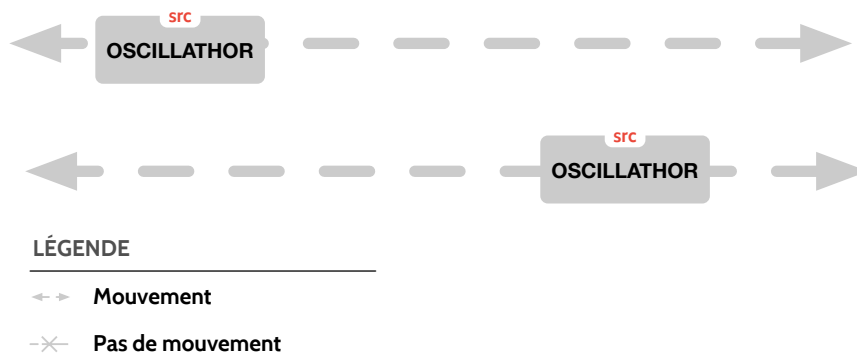


FIGURE 5.6 Battements

### a) Battements

Un premier objet est pris et associé à un comportement *oscillathor*. Dès la création du comportement, on entend le son de cet oscillateur. Un second objet est pris et associé à un autre comportement *oscillathor*. On entend alors deux sons sinusoïdaux dont la fréquence est asservie à la position horizontale des deux objets. Cela permet de reproduire le fameux phénomène de battement qui est entendu quand les fréquences sont trop proches, en déplaçant directement les objets. On entend deux sinusoïdes  $s_1$  et  $s_2$  de périodes  $w_1$  et  $w_2$ . Leurs sons s'additionnent de cette façon :

$$s_1 + s_2 = \cos(w_1 t) + \cos(w_2 t)$$

Le phénomène de battements se produit lorsque les deux fréquences sont trop proches. On entend alors les deux sons comme un seul dont l'amplitude est modulée. On peut expliquer ce phénomène de modulation d'amplitude en réécrivant le son perçu  $s_1 + s_2$  :

$$s_1 + s_2 = 2 \cos\left(\frac{w_1 t + w_2 t}{2}\right) \cos\left(\frac{w_1 t - w_2 t}{2}\right)$$

Cet exemple met en avant la physicalité du contrôle dans les interfaces tangibles musicales. De plus, l'aspect opportuniste peut également être bénéfique pour ce scénario, car l'utilisateur peut se servir d'objets qu'il connaît bien pour découvrir des phénomènes nouveaux (ici, la synthèse sonore).

### b) Ukulélé

On construit ici un petit instrument à cordes pincées, comme un ukulélé. Chaque corde est l'association de deux objets, un sur lequel est attaché un comportement *corde de guitare*, et un autre sur lequel est attaché un comportement *variateur note* relié à cette corde de guitare. On peut *gratter les cordes* grâce à un autre objet attaché à un comportement *marteau* qu'on va déplacer à proximité des objets *corde de guitare*, tandis que les notes jouées par chaque corde peuvent être modulées grâce à des objets *variateur note* qu'on déplace comme un doigt sur le manche d'un ukulélé.

Cet exemple met plusieurs caractéristiques de la construction musicale opportuniste en avant. Tout d'abord, la structure musicale qu'on a créé présume peu de la disposition des objets, et même s'il est possible d'organiser le tout comme sur une guitare, c'est nullement obligatoire. Ensuite, ce scénario peut être réalisé par plusieurs personnes. Le contrôle de chaque *variateur note* peut revenir à une personne. Cette situation sociale de collaboration

FIGURE 5.6

FIGURE 5.7

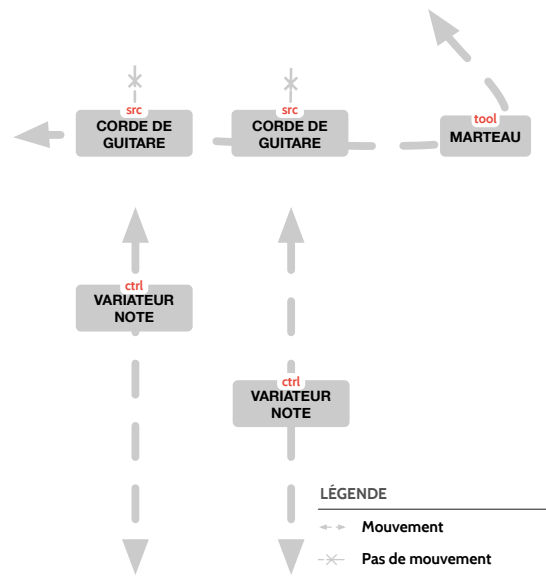


FIGURE 5.7 Ukulélé

colocalisée caractéristique de l'interaction tangible est bien mise en valeur par ce scénario. Enfin, la reconfiguration dynamique peut être également observée, car pendant que deux personnes contrôlent les *cordes de ukulélé* existantes, une troisième peut choisir de rajouter une corde.

### C Séquenceur

FIGURE 5.8

Des *cordes de guitare* sont placées et accompagnées de *variateurs notes*, de façon à pouvoir contrôler quelle note est jouée par chaque corde. On attache ensuite un comportement *tête de lecture* à un objet qui peut être animé d'un mouvement propre, comme par exemple une voiture télécommandée ou un robot du futur (fig. 5.9). Celui-ci, par son déplacement, va *frapper* les cordes de guitare et donc déclencher des sons dans le temps.

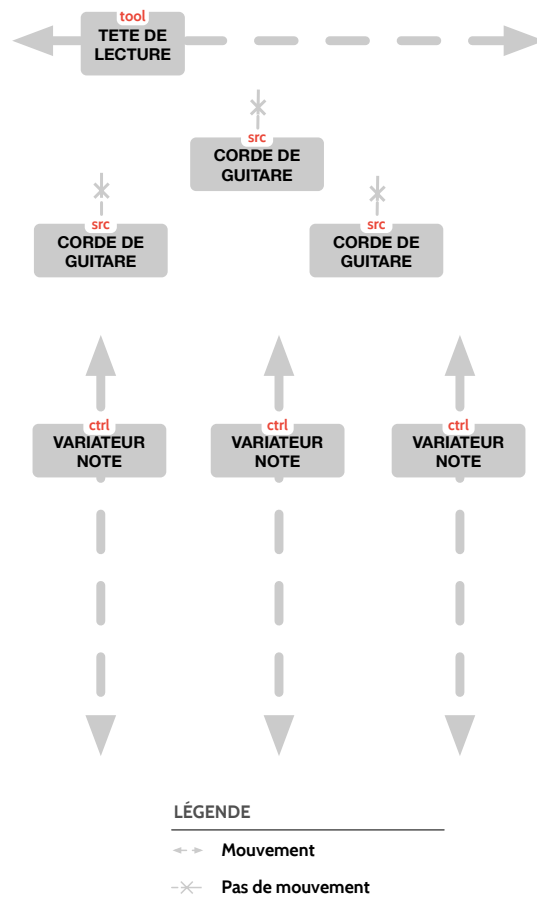
Ce scénario présente un aspect intéressant de l'interaction opportuniste : on peut chercher à profiter de phénomènes naturels ou mécaniques pour générer de la musique. Ici, la structure est celle d'un séquenceur, les événements sont placés dans l'espace et la lecture de ces événements est assujétie à la *tête de lecture*, qui peut se déplacer également de manière non-linéaire. Le temps peut ainsi être accéléré ou ralenti.

## 4 Perspectives

### 1. Limites du modèle actuel

Dans notre prototype actuel, nous nous sommes concentrés sur une architecture de base et les problèmes abordés au chapitre 4. Cependant, un prototype plus complet et plus représentatif de la construction musicale opportuniste peut être développé en intégrant quelques aspects qui ne sont pas abordés dans ce stage :

- **représentation des objets indépendante de la captation** : dans notre prototype, la position des objets est assujétie au système de captation, c'est-à-dire à l'espace filmé

**FIGURE 5.8** Séquenceur**FIGURE 5.9** Un objet doté d'un mouvement propre

par la caméra qui produit les images ou à l'espace dans lequel évoluent les objets repérés avec TUIO, mais une représentation indépendante permettrait une réelle interaction en trois dimensions ;

- **mobilité** : dans notre prototype, on ne prend pas en compte le fait que l'on puisse se déplacer, ce qui rend le théâtre des opérations très limité ;
- **accès concurrents au modèle** : dans notre prototype, on suppose qu'un seul dispositif de captation va modifier le modèle à la fois, mais permettre des accès partagés au système rendrait possible les coopérations entre utilisateurs colocalisés qui évolueraient dans leur environnement ;
- **vraie synthèse modulaire** : nous avons simplifié le modèle de synthèse en limitant les entrées et sorties par processus musical, mais la manipulation d'un vrai système de synthèse modulaire rendrait les possibilités infinies ;
- **comportements partagés** : notre conception associe un comportement à un objet mais l'accent pourrait être porté sur des comportements partagés entre plusieurs objets.

## 2. Prototypage et évaluation des instruments numérique

Une des applications directes de la construction musicale opportuniste, et par la même occasion, de notre prototype, est le prototypage d'instruments numériques. En effet, la construction musicale opportuniste permet de réunir construction et contrôle dans un même espace de manipulation, ce qui a pour effet de rendre immédiatement visibles les résultats des développements qui sont faits. En développant un ensemble cohérent de comportements de plus ou moins haut niveau, il est possible de réaliser des prototypes dans un domaine d'application donné (contrôle de la synthèse, contrôle de structures de haut niveau, contrôle d'événements musicaux, etc.) de manière efficace, car la construction musicale opportuniste réunit construction et test, ce qui permet d'accélérer le processus itératif de prototypage.

Dans l'optique d'une utilisation pour la performance musicale, ou pour l'évaluation de mappings dans le cadre du développement d'instruments de musique numérique, il est important de chercher à intégrer les systèmes de construction musicale opportunistes avec d'autres types d'interfaces plus adaptées au contrôle en temps réel.

## Conclusion

Nous avons exposé les enjeux de l'interaction homme-machine et de l'interaction musicale en temps réel, puis développé spécifiquement ceux de l'interaction tangible pour la musique. Dans ce cadre, nous avons proposé que la construction musicale, étant une part importante de la performance musicale, devait rejoindre le temps et l'espace du contrôle musical, et que l'idée de construction musicale opportuniste permettait de réaliser cette réunion. Après avoir étudié les implications de cette idée sur la conception et la réalisation technique, nous avons isolé trois problèmes auxquels nous avons proposé des solutions. Premièrement, la pauvreté des caractéristiques récupérées dans le suivi par marqueurs peut être enrichie d'une méthode de segmentation. Deuxièmement, les incertitudes, la latence et la discontinuité propres au suivi de position grâce à la vision artificielle ne conviennent pas au contrôle musical, mais l'introduction d'un couplage virtuel rend cette idée possible. Troisièmement, la construction musicale opportuniste doit disposer de techniques d'interaction pour contrôler les étapes de construction, et deux techniques d'affichage de menu sont proposées. Enfin, nous avons proposé un modèle conceptuel pour une première application de la construction musicale opportuniste et détaillé un prototype incarnant ce modèle.

# Bibliographie

- [1] S. ANTIFAKOS, J. BORCHERS et B. SCHIELE : Designing physical interaction with sensor drawbacks in mind. *In. Physical Interaction (PI03) Workshop on Real World User Interfaces*, p. 56. Citeseer, 2003. 21, 29
- [2] D. ARFIB et L. KESSOUS : Gestural control of sound synthesis and processing algorithms. *Gesture and Sign Language in Human-Computer Interaction*, p. 55-85, 2002. 39
- [3] K. BARKATI : *Entre temps réel et temps différé : pratiques, techniques et enjeux de l'informatique dans la musique contemporaine*. Thèse de doctorat, 2009. 13
- [4] M. BEAUDOUIN-LAFON : Instrumental interaction : an interaction model for designing post-wimp user interfaces. *In. Proceedings of the SIGCHI conference on Human factors in computing systems*, p. 446-453. ACM, 2000. 39
- [5] M. BEAUDOUIN-LAFON : Designing interaction, not interfaces. *In. Proceedings of the working conference on Advanced visual interfaces*, p. 15-22. ACM, 2004. 5, 8
- [6] V. BELLOTTI, M. BACK, W. EDWARDS, R. GRINTER, A. HENDERSON et C. LOPES : Making sense of sensing systems : five questions for designers and researchers. *In. Proceedings of the SIGCHI conference on Human factors in computing systems : Changing our world, changing ourselves*, p. 415-422. ACM, 2002. 16
- [7] F. BERGAMASCO, A. ALBARELLI, E. RODOLA et A. TORSSELLO : Rune-tag : a high accuracy fiducial marker with strong occlusion resilience. 2010. 26
- [8] R. BERRY, M. MAKINO, N. HIKAWA et M. SUZUKI : The augmented composer project : The music table. *In. Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality*, p. 338, 2003. 16
- [9] F. BERTHAUT, M. HACHET et M. DESAINTE-CATHERINE : Interaction 3D pour la musique. *2è journées de l'AFRV*, 2007. 11
- [10] M. BILLINGHURST, H. KATO et I. POUPYREV : Tangible augmented reality. *In. ACM SIGGRAPH ASIA 2008*, p. 1-10. ACM, 2008. 17
- [11] A. BLEIWEISS et M. WERMAN : Fusing time-of-flight depth and color for real-time segmentation and tracking. *Dynamic 3D Imaging*, p. 58-69, 2009. 29
- [12] J. BOWERS et N. VILLAR : Creating ad hoc instruments with Pin&Play&Perform. *In. Proceedings of the 2006 conference on New interfaces for musical expression*, p. 234-239, 2006. 14, 17
- [13] D. BOWMAN, E. KRUIJFF, J. LAVIOLA et I. POUPYREV : *3D user interfaces : theory and practice*, vol. 2. Addison-Wesley Boston (MA), 2005. 32
- [14] C. CADOZ et M. WANDERLEY : Gesture-music. *Trends in Gestural Control of Music*, p. 71-94, 2000. 10
- [15] R. CHALON : *Réalité mixte et travail collaboratif : IRVO, un modèle de l'interaction homme-machine*. Thèse de doctorat, 2004. 7

- [16] B. CHEN, K. CHENG, H. CHU, S. KUO, R. LIANG, M. YU, R. LIANG, H. LIN et Y. CHU : MemoICON : using everyday objects as physical icons. *In. ACM SIGGRAPH ASIA 2009 Art Gallery & Emerging Technologies : Adaptation*, p. 78--78. ACM, 2009. 17, 18
- [17] J. COLGATE et G. SCHENKEL : Passivity of a class of sampled-data systems : Application to haptic interfaces. *In. American Control Conference, 1994*, vol. 3, p. 3236--3240. IEEE, 1994. 30
- [18] J. COLGATE, M. STANLEY et J. BROWN : Issues in the haptic display of tool use. *In. Proceedings of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, p. 140--145. IEEE, 1995. 30
- [19] N. COLLINS, A. MCLEAN, J. ROHRHUBER et A. WARD : Live coding in laptop performance. *Organised Sound*, 8(03):321--330, 2003. 13
- [20] E. COSTANZA, S. SHELLEY et J. ROBINSON : Introducing audio d-touch : A tangible user interface for music composition and performance. *In. Proceedings of the 6th International Conference on Digital Audio Effects*, p. 63--70, 2003. 26
- [21] J.-M. COUTURIER : Espaces interactifs visuels pour le contrôle des sons musicaux. *In. Journées d'études Espaces Sonores*, 2002. 11
- [22] J.-M. COUTURIER : A model for graphical interaction applied to gestural control of sound. *In. Proceedings of the international conference on Sound and Music Computing*, 2006. 11, 39
- [23] R. DACHSELT et A. HUBNER : Three-dimensional menus : A survey and taxonomy. *Computers & Graphics*, 31(1):53--65, 2007. 32
- [24] W. DANIEL et S. DIETER : ARToolKitPlus for pose tracking on mobile devices. *In. Proceedings of the 12th Computer Vision Winter Workshop*, 2007. 25, 26
- [25] P. DAVIDSON et J. HAN : Synthesis and control on large scale multi-touch sensing displays. *In. Proceedings of the 2006 conference on New interfaces for musical expression*, p. 216--219, 2006. 14
- [26] P. DOURISH : Embodied interaction : Exploring the foundations of a new approach to HCI. 1999. 8
- [27] P. DOURISH : *Where the action is : the foundations of embodied interaction*. The MIT Press, 2004. 8
- [28] A. ELGAMMAL, R. DURAI SWAMI et L. DAVIS : Efficient kernel density estimation using the fast gauss transform with applications to color modeling and tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 1499--1504, 2003. 27
- [29] Y. FERNAEUS, J. THOLANDER et M. JONSSON : Beyond representations : towards an action-centric perspective on tangible interaction. *International Journal of Arts and Technology*, 1(3):249--267, 2008. 8
- [30] Y. FERNAEUS, J. THOLANDER et M. JONSSON : Towards a new set of ideals : consequences of the practice turn in tangible interaction. *In. Proceedings of the 2nd international conference on Tangible and embedded interaction*, p. 223--230. ACM, 2008. 8
- [31] M. FIALA : Artag, a fiducial marker system using digital techniques. *In. Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, p. 590--596. IEEE Computer Society, 2005. 26
- [32] T. FISCHER et W. LAU : Marble track music sequencers for children. *In. Proceedings of the 2006 conference on Interaction design and children*, p. 141--144. ACM, 2006. 11
- [33] K. FISHKIN : A taxonomy for and analysis of tangible interfaces. *Personal and Ubiquitous Computing*, 8(5):347--358, 2004. 7

- [34] G. FITZMAURICE : *Graspable user interfaces*. Thèse de doctorat, University of Toronto, 1996. 6
- [35] P. FUA : A parallel stereo algorithm that produces dense depth maps and preserves image features. *Machine Vision and Applications*, 6(1):35–49, 1993. 25
- [36] E. GHOMI : Utilisation de modèles intermédiaires pour le mapping de paramètres de synthèse. Mémoire de D.E.A., Université Pierre et Marie Curie, 2008. 39
- [37] T. GRILL : dyn : Dynamic Object Management. *In. Proceedings of the 1st International pd Convention*, 2004. 40
- [38] J. GRUDIN : A moving target : The evolution of human-computer interaction. *The human-computer interaction handbook : fundamentals, evolving technologies, and emerging applications*, 2003. 4
- [39] M. HACHET, A. KIAN, F. BERTHAUT, J.-S. FRANCO et M. DESAINTE-CATHERINE : Opportunistic music. *In. JVRG 2009 (EGVE - ICAT - EuroVR)*, p. 45–51, 2009. 17
- [40] S. HENDERSON et S. FEINER : Opportunistic controls : leveraging natural affordances as tangible user interfaces for augmented reality. *In. Proceedings of the 2008 ACM symposium on Virtual reality software and technology*, p. 211–218. ACM, 2008. 17
- [41] S. HENDERSON et S. FEINER : Opportunistic tangible user interfaces for augmented reality. *IEEE Transactions on Visualization and Computer Graphics*, 16(1):4–16, 2010. 17
- [42] A. HUNT, M. WANDERLEY et M. PARADIS : The importance of parameter mapping in electronic instrument design. *Journal of New Music Research*, 32(4):429–440, 2003. 39
- [43] H. ISHII : Tangible bits : beyond pixels. *In. Proceedings of the 2nd international conference on Tangible and embedded interaction*, p. xv–xxv. ACM, 2008. 15
- [44] H. ISHII et B. ULLMER : Tangible bits : towards seamless interfaces between people, bits and atoms. *In. Proceedings of the SIGCHI conference on Human factors in computing systems*, p. 234–241. ACM, 1997. 6
- [45] R. JACOB, A. GIROUARD, L. HIRSHFIELD, M. HORN, O. SHAER, E. SOLOVEY et J. ZIGELBAUM : Reality-based interaction : a framework for post-WIMP interfaces. *In. Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, p. 201–210. ACM, 2008. 7
- [46] B. JONES, R. SODHI, R. CAMPBELL, G. GARNETT et B. BAILEY : Build your world and play in it : Interacting with surface particles on complex objects. *In. 9th IEEE International Symposium on Mixed and Augmented Reality*, p. 165–174. IEEE, 2010. 17, 18
- [47] S. JORDÀ, G. GEIGER, M. ALONSO et M. KALTENBRUNNER : The reacTable : exploring the synergy between live music performance and tabletop tangible interfaces. *In. Proceedings of the 1st international conference on Tangible and embedded interaction*, p. 139–146. ACM, 2007. 11, 15
- [48] S. JORDÀ, M. KALTENBRUNNER, G. GEIGER et R. BENCINA : The reacTable\*. *In. Proceedings of the International Computer Music Conference*, p. 579–582, 2005. 11, 15, 17
- [49] M. KALTENBRUNNER et R. BENCINA : reactivation : a computer-vision framework for table-based tangible interaction. *In. Proceedings of the 1st international conference on Tangible and embedded interaction*, p. 69–74. ACM, 2007. 25, 26
- [50] M. KALTENBRUNNER, T. BOVERMANN, R. BENCINA et E. COSTANZA : Tuio : A protocol for table-top tangible user interfaces. *In. Proc. of the The 6th Int'l Workshop on Gesture in Human-Computer Interaction and Simulation*. Citeseer, 2005. 42

- [51] M. KALTENBRUNNER, G. GEIGER et S. JORDÀ : Dynamic patches for live musical performance. *In. Proceedings of the 2004 conference on New interfaces for musical expression*, p. 19–22, 2004. 14, 38
- [52] K. KARPLUS et A. STRONG : Digital synthesis of plucked-string and drum timbres. *Computer Music Journal*, 7(2):43–55, 1983. 39
- [53] H. KATO et M. BILLINGHURST : Marker tracking and HMD calibration for a video-based augmented reality conferencing system. *In. Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality*, p. 85. IEEE Computer Society, 1999. 25
- [54] A. KIAN : Contrôle opportuniste de la musique. Mémoire de D.E.A., Université Bordeaux I, 2009. 3, 39
- [55] V. LEPETIT et P. FUA : *Monocular model-based 3D tracking of rigid objects*. Now Publishers Inc, 2005. 23
- [56] Y. MANN, J. LUBOW et A. FREED : The tactus : a tangible, rhythmic grid interface using found-objects. *In. Proceedings of the 2009 conference on New interfaces for musical expression*, 2009. 17
- [57] D. MERRILL et H. RAFFLE : The sound of touch. *In. CHI'07 extended abstracts on Human factors in computing systems*, p. 2807–2812. ACM, 2007. 15
- [58] P. MILGRAM et F. KISHINO : A taxonomy of mixed reality visual displays. *IEICE Transactions on Information and Systems*, 77:1321–1321, 1994. 7
- [59] V. I. MORARIU, B. V. SRINIVASAN, V. C. RAYKAR, R. DURAISWAMI et L. S. DAVIS : Automatic online tuning for fast gaussian summation. *In. Advances in Neural Information Processing Systems*, 2008. 27
- [60] T. MÖRWALD, J. PRANKL, A. RICHTSFELD, M. ZILlich et M. VINCZE : Blort - the blocks world robotic vision toolbox. *In. Proc. ICRA Workshop Best Practice in 3D Perception and Modeling for Mobile Manipulation*, 2010. 23
- [61] H. NEWTON-DUNN, H. NAKANO et J. GIBSON : Block jam : a tangible interface for interactive music. *Journal of New Music Research*, 32(4):383–393, 2003. 11
- [62] H. NISHINO : Topolo Surface : A 2D fiducial tracking system based on topological region adjacency and angle information. *Information and Media Technologies*, 5(2):479–488, 2010. 26
- [63] D. NORMAN : *The design of everyday things*. Basic Books New York, 2002. 16
- [64] E. PARVIZI et Q. WU : Real-time approach for adaptive object segmentation in time-of-flight sensors. *In. 20th IEEE International Conference on Tools with Artificial Intelligence*, p. 236–240. IEEE, 2008. 26
- [65] J. PATTEN et H. ISHII : A comparison of spatial organization strategies in graphical and tangible user interfaces. *In. Proceedings of DARE 2000 on Designing augmented reality environments*, p. 41–50. ACM, 2000. 6
- [66] J. PATTEN, H. ISHII, J. HINES et G. PANGARO : Sensetable : a wireless object tracking platform for tangible user interfaces. *In. Proceedings of the SIGCHI conference on Human factors in computing systems*, p. 253–260. ACM, 2001. 22
- [67] J. PATTEN, B. RECHT et H. ISHII : Audiopad : a tag-based interface for musical performance. *In. Proceedings of the 2002 conference on New interfaces for musical expression*, p. 1–6, 2002. 11
- [68] J. PATTEN, B. RECHT et H. ISHII : Interaction techniques for musical performance with tabletop tangible interfaces. *In. Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology*, p. 27–es. ACM, 2006. 32

- [69] E. PEDERSEN et K. HORNBAEK : mixiTUI : a tangible sequencer for electronic live performances. *In. Proceedings of the 3rd International Conference on Tangible and Embedded Interaction*, p. 223–230. ACM, 2009. 11
- [70] H. PENG : TouchSound : making sounds with everyday objects. *In. Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction*, p. 439–440. ACM, 2011. 11, 23
- [71] S. PERREAULT et P. HÉBERT : Median filtering in constant time. *IEEE Transactions on Image Processing*, 16(9):2389–2394, 2007. 27
- [72] I. POUPYREV, D. TAN, M. BILLINGHURST, H. KATO, H. REGENBRECHT et N. TETSUTANI : Developing a generic augmented-reality interface. *Computer*, 35(3):44–50, 2002. 16, 19, 32
- [73] M. PUCKETTE : Pure data. *In. Proc. International Computer Music Conference, 1996*. ICMA, 1996. 40
- [74] V. RAYKAR, R. DURAISWAMI et L. ZHAO : Fast computation of kernel estimators. *Journal of Computational and Graphical Statistics*, 19(D):205–220, 2010. 27
- [75] J. REKIMOTO et Y. AYATSUKA : Cybercode : designing augmented reality environments with visual tags. *In. Proceedings of DARE 2000 on Designing augmented reality environments*, p. 1–10. ACM, 2000. 25
- [76] J. REKIMOTO et K. NAGAO : The world through the computer : Computer augmented interaction with real world environments. *In. Proceedings of UIST*, vol. 95, p. 29–36, 1995. 5, 25
- [77] J. SCOTT et B. DRAGOVIC : Audio location : Accurate low-cost location sensing. *Pervasive Computing*, p. 1–18, 2005. 22
- [78] B. ULLMER et H. ISHII : Emerging frameworks for tangible user interfaces. *IBM Systems Journal*, 39(3/4):915–931, 2000. 7, 15
- [79] E. van den HOVEN et B. EGGEN : Tangible computing in everyday life : Extending current frameworks for tangible user interfaces with personal objects. *In. Proceedings of EUSAI 2004*, p. 230–242. Springer, 2004. 17, 18
- [80] G. WANG, A. MISRA, A. KAPUR et P. COOK : Yeah, ChucK it! ⇒ dynamic, controllable interface mapping. *In. Proceedings of the 2005 conference on New interfaces for musical expression*, p. 196–199, 2005. 14
- [81] Y. WANG : Image segmentation based on parameter estimation. *International archives of photogrammetry and remote sensing*, 29:600–606, 1993. 25
- [82] P. WEGNER : Why interaction is more powerful than algorithms. *Communications of the ACM*, 40(5):80–91, 1997. 4
- [83] G. WEINBERG et S. GAN : The squeezables : Toward an expressive and interdependent multi-player musical instrument. *Computer Music Journal*, 25(2):37–45, 2001. 11
- [84] M. WEISER : Ubiquitous computing. *Computer*, 26(10):71–72, 1993. 5
- [85] D. WESSEL et M. WRIGHT : Problems and prospects for intimate musical control of computers. *Computer Music Journal*, 26(3):11–22, 2002. 24, 29
- [86] S. WHITE, D. FENG et S. FEINER : Interaction and presentation techniques for shake menus in tangible augmented reality. *In. Proceedings of 2009 IEEE International Symposium on Mixed and Augmented Reality*, p. 39–48, 2009. 32, 33
- [87] M. WRIGHT et A. FREED : Open sound control : A new protocol for communicating with sound synthesizers. *In. Proceedings of the 1997 International Computer Music Conference*, p. 101–104, 1997. 40

- [88] R. ZIOLA, S. GRAMPUROHIT, N. LANDES et J. FOGARTY : OASIS : Examining a framework for interacting with general-purpose object recognition. Rap. tech., Intel Labs Seattle, may 2010. 23