

# Compte-rendu sur les méthodes spatiales dans le projet OSSIA.

## 1 Préambule

Ce document vise à présenter les études qui ont été menées en lien avec la manipulation de données spatiales dans le cadre du projet OSSIA.

Il existe trois défis majeurs : la conception d'un modèle de données générique et assez puissant pour couvrir les cas que nous allons présenter, la représentation graphique de ces données lorsqu'un grand nombre de contrôles sont présents, et le contrôle et l'édition par l'utilisateur.

Nous analyserons plusieurs types de projets différents dans lesquels une notion de contrôle spatial existe, et qui sont utiles à la spécification des besoins.

Puis, nous présenterons les solutions et prototypes qui ont été développés dans le cadre du projet OSSIA, afin de tenter de répondre à au moins partie des besoins qui ont été exprimés.

## 2 Exemples

Les exemples sont répartis en trois catégories : ceux liés aux problématiques spatiales dans un contexte sonore et musical, ceux liés aux conceptions d'interfaces, et un cas de contrôle robotique.

### 2.1 Cadre sonore

La spatialisation en son, puis musique est une pratique très courante et une recherche importante a eu lieu sur ce sujet.

Il existe deux problèmes distincts que l'on place parfois sous le même terme de spatialisation du son :

- La composition du son avec des composantes spatiales : conception de trajectoires qui vont faire tourner le son, etc.<sup>1</sup>
- Les méthodes et algorithmes permettant le rendu de cet espace avec un ensemble de hauts-parleurs ou au casque (HRTF, etc.).

C'est le premier point qui nous intéresse ici. Nous pouvons le préciser encore plus en étudiant les fonctionnements et conceptions différentes qu'il peut y avoir selon que l'objectif soit de composer de la musique, ou bien de concevoir des systèmes interactifs.

---

1. *Vocabulaire de l'espace en musiques électroacoustiques*, 2006, Bertrand Merlier, et *La spatialisation des musiques électro-acoustiques*. 2012, Laurent Pottier.

### 2.1.1 Trajectoires sonores

Dans un cadre musical, on aura tendance à composer par pistes, et donc à vouloir spatialiser piste par piste. Des objets sonores seront disposés dans un séquenceur tel que Pro Tools ou Reaper, puis, au moment de la composition ou bien lors d'une interprétation en concert, les différentes pistes sont envoyées sur différents canaux audio.

Il existe des méthodes plus sophistiquées permettant de définir des trajectoires graphiquement, comme IanniX en figure1, ou bien à l'aide de modèles inspirés de lois physiques, auquel cas des applications spécifiques sont développées pour la composition. Par exemple, les modèles masse-ressort sont souvent utilisés dans ce but.

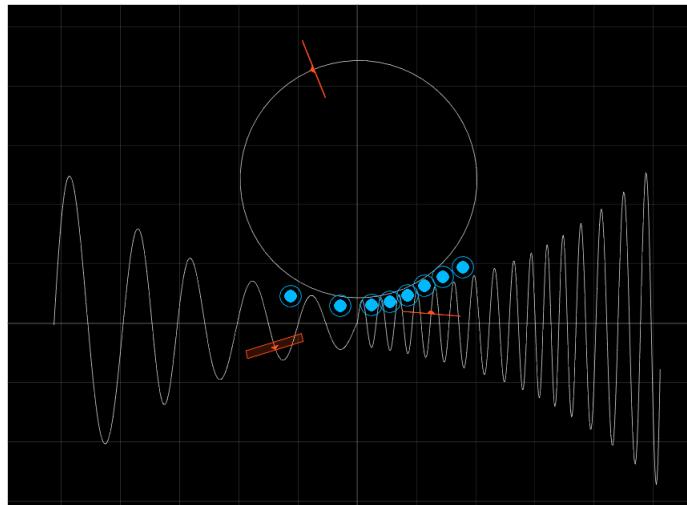


FIGURE 1 – Une partition composée de trajectoires et de triggers dans le séquenceur graphique IanniX

### 2.1.2 Positionnement d'objets sonores

Dans un cadre muséographique, on a plus souvent des objets sonores qui peuvent être associés à des éléments graphiques : des notes dans un manuscrit, un personnage sur un tableau... On désire alors contrôler le comportement de cet objet sonore au fil du temps et le scénariser, le faire se déplacer, changer ses propriétés sonores, etc.

Cette approche est aussi plus pertinente dans un cadre vidéo-ludique : en effet, les outils de conception de jeu-vidéo (*citer partie rapport cnam*) ont une approche fortement orientée objet - la méthode actuellement la plus répandue dans l'industrie est le pattern Entity-Component-Systems<sup>2</sup>, qui permet d'encapsuler différentes couches applicatives (qui peuvent contenir des données, des comportements...) très simplement dans n'importe quel objet du jeu.

Une application a été développée par Blue Yeti : **Sonoplui**e. Ici, des utilisateurs font un parcours dans une ville. Ils ont un casque et sont géolocalisés. En fonction de leur position et de leur parcours, ils entendent des messages rappelant des évènements historiques de la ville.

De plus, si des utilisateurs se croisent, l'application peut en avoir conscience et déclencher des évènements spécifiques.

Cette application a plusieurs pré-requis :

2. Component Based Engine Design, Randy Gaul : <http://www.randygaul.net/2013/05/20/component-based-engine-design/>

- L'outil de conception doit, au minimum, permettre de placer des points sur une carte et de définir une action sur le périphérique de l'utilisateur, lorsqu'il rencontre un point.
- Pour plus de richesse, des zones sonores peuvent être définies : par exemple, une bande de plage.
- Il doit y avoir une notion de scénarisation telle qu'offerte par i-score, permet de construire une histoire au cours du temps, plutôt que de simplement avoir une liste de hot-spots interactifs.
- Enfin, se pose la question de la répartition et de l'interaction entre plusieurs protagonistes. Une possibilité serait de définir des zones d'interactions qui ne s'activent que lorsque plusieurs participants sont présents à l'intérieur. Les périphériques des utilisateurs présents dans cette zone doivent communiquer entre eux (ou bien communiquer via un serveur, mais cela nécessite des appareils possédant une connection internet).

## 2.2 Interactivité et interfaces

Ici, nous nous intéressons aux possibilités de création et de contrôle dynamique d'interfaces graphiques. La plupart des paradigmes d'UI permettent de définir des vues dans un éditeur graphique, mais ces vues sont en général statiques et il est nécessaire de recourir à des langages de programmation pour avoir une évolution personnalisable dans le logiciel.

Nous aborderons ici deux cas : le premier est celui des interfaces utilisateur programmables graphiquement, ce qui peut s'avérer utile quand il y a de nombreuses animations et objets graphiques non-usuels. Le second est celui des interfaces nécessaires pour contrôler des espaces de paramètres particuliers.

Il reste des problèmes non abordés : notamment la création d'objets dynamique, la visualisation, et la gestion de comportements d'objets en masse, comme par exemple pour un générateur de particules.

### 2.2.1 Contrôle d'interfaces utilisateurs

Le premier cas est celui d'une application en mode kiosque, possédant un menu de choix des langues, qui mène pour chaque langue à un menu principal permettant de choisir des vidéos à lire en mosaïque.

Lorsqu'une langue est choisie, l'écran principal glisse depuis la droite. Les boutons ont une animation lorsqu'ils sont pressés. Lorsqu'une vidéo est pressée, elle se met en lecture. Des *gestures* permettent de déplacer les vidéos, en suivant des lois de physique basiques (notamment, les vidéos peuvent tourner et rentrer en collision).

Les problématiques, ici, sont :

- Le contrôle d'un grand nombre d'éléments statiques, et la hiérarchisation de ces éléments. Par exemple, si les boutons de langues sont enfants du premier menu, on veut qu'ils suivent le déplacement de leur menu parent.
- Un questionnement sur les modèles physiques. Cela a déjà été abordé pour les modèles masse-ressort évoqués en spatialisation de musique.
- La gestion d'animations. Généralement, une animation consiste en l'interpolation de paramètres dans le temps, en passant par des états prédéfinis (keyframes).

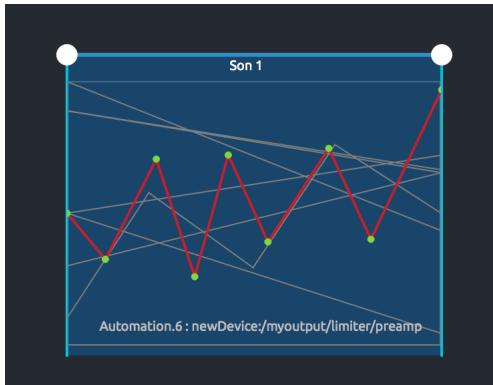


FIGURE 2 – De nombreux paramètres sur un seul axe dans i-score, rendant la partition peu lisible.

### 2.2.2 Navigation dans espaces de paramètres

Un problème plus général est celui de la gestion des espaces de paramètres particuliers.

Par exemple, pour choisir et animer la couleur d'un élément dans le temps, il est nécessaire d'avoir des outils de création adaptés, tout en gardant les primitives de trajectoires et d'objets évoquées précédemment.

Un second point est celui de la gestion des correspondances entre espaces de données. Ainsi, si on désire automatiser un point sur un plan en deux dimensions dans le temps, il doit être possible de le faire selon les systèmes de coordonnées cartésiennes ou polaires.

De même, pour les couleurs il est important de pouvoir travailler en RVB, mais aussi en L\*a\*b\*, HSL, et notamment de pouvoir passer facilement d'un mode à l'autre.

Enfin, le dernier point est le cas d'une aide à l'écriture pour une application comportant un grand nombre de paramètres, potentiellement artistiques. Le problème de la visualisation se pose lorsque ces paramètres sont représentés sur une seule dimension, comme en fig. 2.

## 2.3 L'espace pour le spectacle

### 2.3.1 Danse interactive

Dans cette application, nous désirons travailler avec plusieurs danseurs sur scène, et produire un comportement émergent au gré de leurs déplacements et du scénario du spectacle.

Le premier cas est très simple : il existe un cercle virtuel et invisible au sol, délimité par l'auteur du spectacle, et on désire allumer une lumière lorsqu'un danseur rentre dans le cercle. C'est similaire au cas de Sonoplui vu plus tôt et se rapporte à un simple test booléen.

Nous aimerais maintenant créer de la géométrie. Soient deux danseurs : la seconde partie du spectacle consiste à allumer la lumière lorsque le barycentre des deux danseurs est dans le cercle virtuel. Cela requiert, en plus de ce qui a été dit précédemment, de la nécessité d'avoir une géométrie dynamique et définie par des paramètres provenant de l'extérieur.

Enfin, dans la dernière partie du spectacle, nous utilisons la distance entre les deux danseurs pour faire varier le rayon du cercle au centre de la scène.

Dans cet exemple, nous avons donc comme caractéristique principale une grande liberté de contrôle laissée à l'auteur, ainsi que la capacité de définir des géométries complexes.

### 2.3.2 Contrôle de robots

Une application intéressante est celle des chorégraphies de robot. Ce chantier, commencé en 2015, vise à étendre les possibilités du séquenceur i-score d'abord à des robots quadrupèdes Metabots (fig. 3), puis, potentiellement à des drones.

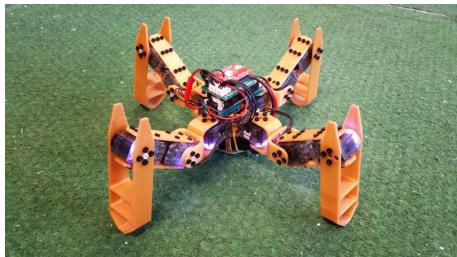


FIGURE 3 – Un metabot.

Là encore, la question de l'écriture se pose, et notamment le cas de la gestion d'un grand nombre d'objets dans un espace relativement restreint.

De plus, ces objets ne vont pas forcément réagir comme prévu : si par exemple un robot est programmé pour aller tout droit, mais qu'il rencontre un autre robot qui serait tombé en panne sur le chemin, actuellement le premier robot ne saurait en tenir compte et rentrerait en collision avec le second robot.

## 3 Réalisations

Les réalisations se séparent en deux catégories : la première consiste en des extensions pour i-score ou d'autres logiciels et frameworks. Ces logiciels ont été retenus car ils sont directement en lien avec les problématiques qui ont été évoquées précédemment.

Puis, une fois ces cas étudiés, nous avons tenté d'apporter une solution originale au problème d'écriture qui se pose.

### 3.1 Clients manipulant des données spatiales

#### 3.1.1 Interopérabilité avec Qt

Un plug-in pour i-score permettant d'exposer l'arbre d'objets d'une application Qt a été développé. Cela permet de contrôler facilement la position et le contenu des éléments d'une interface graphique réalisée avec cette technologie depuis i-score.

#### 3.1.2 Contrôle d'OpenAL

OpenAL est une bibliothèque ayant été développée pour les besoins des jeux vidéos au début des années 2000. Elle se base sur une notion de sources sonores associées à des fichiers sons, qui sont positionnées en trois dimensions, ainsi que d'un listener qui se déplace et perçoit donc les sons plus ou moins forts. Une implémentation récente, **OpenAL-soft**<sup>3</sup>, permet d'utiliser des HRTF pour avoir une sensation de spatialisation réaliste au casque.

3. <http://kcat.strangesoft.net/openal.html>

Un outil exposant des sources OpenAL via le protocole Minuit et permettant de charger des sons, puis de les positionner dans l'espace a été implémenté. Ces sources sont donc contrôlables depuis i-score.

### 3.1.3 Contrôle de metabots

Une couche de communication entre i-score et les robots Metabots a été développée et a mené à une présentation aux Robot Makers Day, avec notamment une chorégraphie où deux robots sont contrôlés et synchronisés musicalement avec i-score.

Cela a été l'occasion de comparer le développement d'une chorégraphie entre i-score, et un langage de programmation graphique inspiré de Scratch. Les résultats ont été probants : il a fallu beaucoup moins de temps pour produire une même chorégraphie dans i-score que dans l'autre langage.

Ce travail continue : un groupe d'étudiants d'école d'ingénieur travaille actuellement à des chorégraphies avec un nombre élevé de Metabots, ainsi qu'au travail avec des drones. Cela nécessite notamment la conception d'outils de simulation appropriés.

## 3.2 État de l'art sur méthodes spatiales, positionnement, trajectoires

### 3.2.1 Free-hand drawing sur tableau

À titre d'exemple pour des démonstrations et exposés, un outil permettant de dessiner des zones de manières arbitraires sur un tableau a été développé (cf. fig. 4).

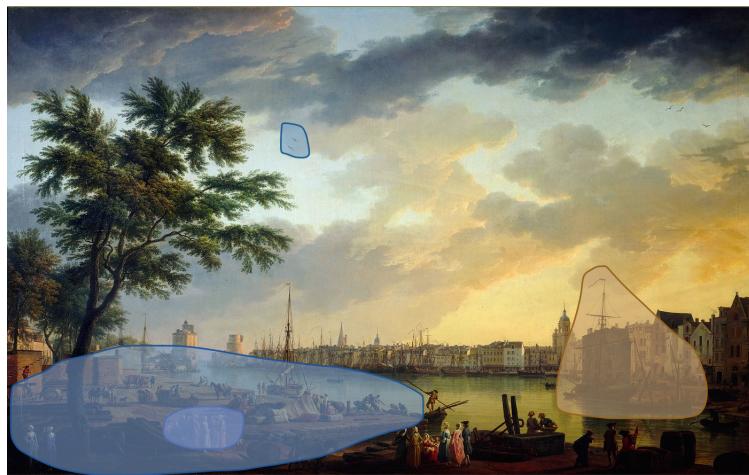


FIGURE 4 – Un tableau sur lequel on peut apposer des couches sonores.

Lorsqu'une zone est activée, elle produit un son qui peut être en relation avec les éléments sous-jacents.

### 3.2.2 Liste des tâches et prototype pour processus de mapping spatial

Suite aux différents cas d'utilisation qui ont été présentés, un état de l'art, présent en annexe B, a été effectué dans le domaine de l'authoring spatial, et des techniques et méthodes possibles pour la modélisation et représentation de telles données.

Cet état de l'art a ensuite conduit à la recherche de méthodes possibles pour combler le besoin existant pour un outil permettant de spécifier des scénarios manipulant beaucoup de données spatiales entrelacées entre elles, ainsi qu'avec une évolution temporelle.

Enfin, une liste des tâches pour une première implémentation dans i-score a été donnée. Elle est présente en annexe A. Le premier point auquel nous nous intéressons est celui du modèle de données. Il existe deux courants : les approches orientées langage, et les approches orientées données.

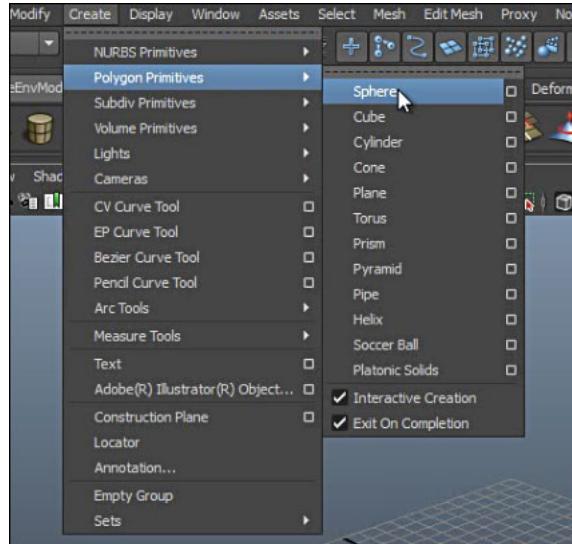


FIGURE 5 – Primitives graphiques dans le logiciel de 3D Maya.

Les premières sont des langages de programmation spécifiquement étudiés pour résoudre le problème posé de manière élégante. Plusieurs possibilités ont été étudiées :

- Utiliser un langage de script existant, tel que Lua, Python ou JavaScript pour définir les objets et comportements spatiaux. L'avantage est une très grande flexibilité, mais les capacités d'analyse statique et d'optimisation sont diminuées si l'intégralité du langage est accessible à l'utilisateur.
- Il existe une famille de langages plus spécialisés dans la gestion de données interactives : c'est le courant de la programmation fonctionnelle réactive<sup>4</sup>.
- Un DSL (langage spécifique à un domaine) pensé pour la création de graphismes vectoriels est présent dans le package Haskell **DIAGRAMS**. Il fonctionne à l'aide d'un jeu de primitives réduites qui doivent être composées par la suite.
- L'approche que nous avons pour l'instant retenue est d'utiliser un système de calcul symbolique<sup>5</sup>. Ainsi, le langage utilisé est simple : ce sont les mathématiques traditionnelles. L'avantage est la relative simplicité à l'utilisation par rapport à la nécessité d'apprendre un langage complet, ainsi que la possibilité de changer de moteur de résolution si celui choisi n'offre pas les performances adéquates. En revanche, il n'est plus possible de gérer des cas comme des boucles, etc.

En parallèle, les approches orientées données sont souvent plus adaptées à une manipulation directe par un utilisateur novice, car elles sont basées sur des outils graphiques, ou de plus haut niveau :

- Il existe des modèles qualitatifs de représentation des données spatiales : RCC-8, OPRA-4, etc. Ces modèles sont principalement étudiés dans le monde de l'intelligence artificielle et de l'ingénierie des connaissances. Ils sont notamment utilisés pour la reconnaissance d'images, de lieux, etc.

4. *Arrows, robots, and functional reactive programming*, 2003, Paul Hudak

5. En pratique le choix s'est porté sur GiNaC : [www.ginac.de](http://www.ginac.de)

- Le parc de logiciels graphiques existant se base souvent sur quelques primitives simples qui sont ensuite composées entre elles, ainsi que sur l'utilisation de modèles 3D, comme pour Maya en figure 5.

La méthode que nous proposons combine les deux : utiliser un Computer Algebra System (CAS) pour spécifier des objets de base avec lesquels travailler, puis permettre à l'utilisateur de faire des transformations graphiques - translation, rotation, scale - sur les objets. Les paramètres de ces objets sont extraits de la formule mathématique et séparés en deux ensembles : ceux qui font partie de l'espace d'évaluation et ceux qui font partie de l'espace de paramétrisation. Par la suite, des propriétés peuvent être extraites à l'aide d'algorithmes intervenant plus tard, comme la collision entre deux objets. Ces propriétés sont ensuite utilisables dans le reste d'i-score via l'arbre de paramètres.

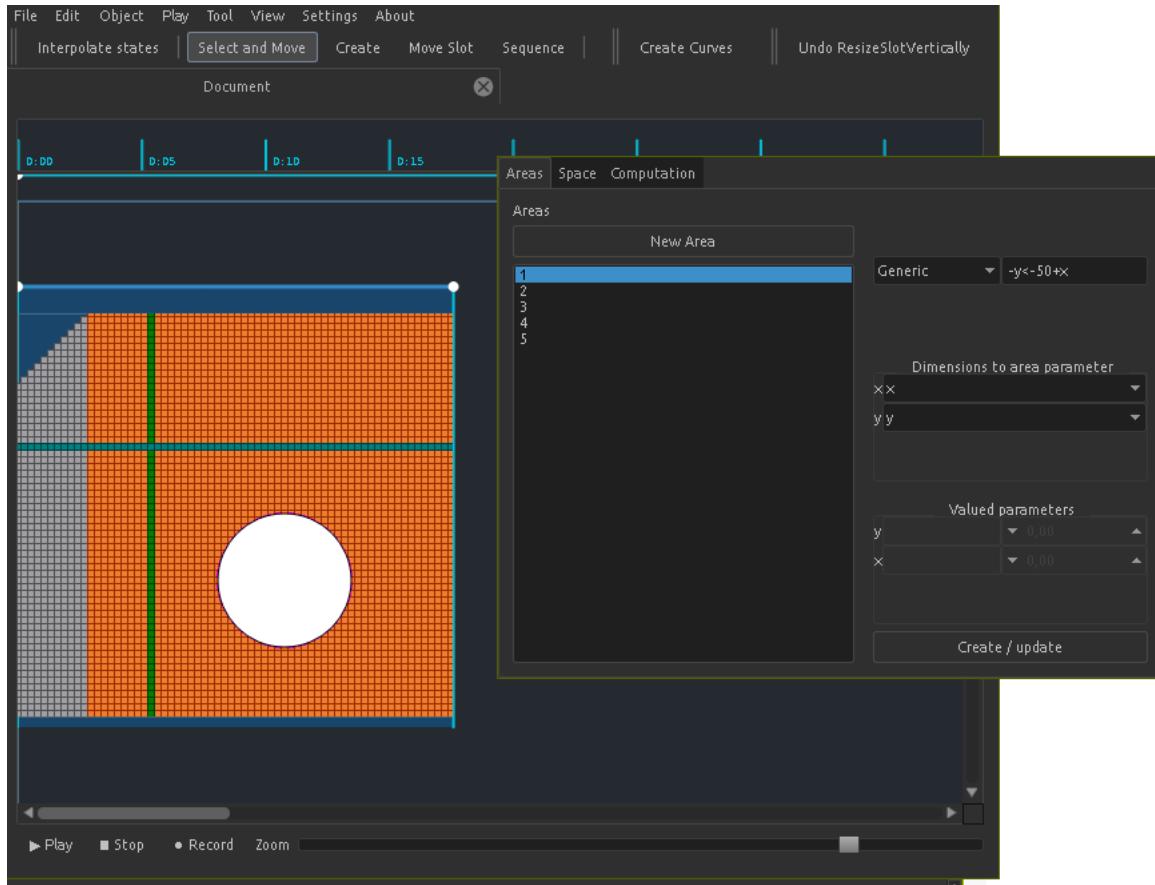


FIGURE 6 – Processus spatial dans i-score, ainsi que l'interface d'édition.

Le rendu se fait actuellement par un algorithme naïf de pixelisation, qui devra par la suite être étendu à de la voxelisation pour une manipulation en trois dimensions. Pour des objets définis dans plus de trois dimensions, il est possible de choisir la restriction que l'on visualise. Un exemple est donné en figure 6, où cinq zones sont représentées :

- 1 :  $-y < -50 - x$  (en gris).
  - 2 :  $x > 50$  (en orange).
  - 3 :  $x == 100$  (en bleu).
  - 4 :  $y == 100$  (en vert).
  - 5 :  $(x - x_0)^2 + (y - y_0)^2 \leq r^2$  pour  $x_0 = 200, y_0 = 200, r = 50$  (en blanc). Comme nous savons qu'il s'agit d'un cercle, il est possible d'offrir une meilleure représentation.
- La manipulation et l'édition des objets ainsi créés est faite via l'édition de l'équation de

manière textuelle, et nous désirons étudier les interactions utilisateur possibles pour une édition via des modalités purement graphique. Il existe aussi des bibliothèques permettant de meilleurs rendus, comme VTK<sup>6</sup> qui gère la géométrie paramétrique.

---

6. [www.vtk.org](http://www.vtk.org)

# **Annexes**

## **A   Liste de tâches pour le développement d'un processus spatial dans i-score.**

# Space

Printed from Asana

- Sérialisation
  - Sauvegarde (DataStream)
  - Sauvegarde (JSON)
  - Lecture (DataStream)
  - Lecture (JSON)
- Mise à jour de l'ui
  - D'où proviennent les signaux ?
  - Activer listen spécifiquement par zone ?
  - Fixer valeur par défaut ?
- Algorithmes
  - Tangent
  - Overlap
  - Equals
  - !Equals
- Approximations
  - Format simple (cubique)
  - Voir ce qu'on peut faire avec des quadtree, des laplace, etc...
  - Essayer plusieurs algos de parcours
  - Méthodes d'approx. numérique ? (newton-raphson, etc..)
  -
- Gestion du cache
  - Comparaison entre caches approximés ?
- Gérer les bornes correctement
  - Bornes de l'espace
  - Bornes par zone
    - Sur les paramètres
    - Sur les variables d'espace?
- Modèles spécialisés
  - Cercle
    - position x,y et rayon r
  - Pointeur
  - Rectangle
    - centre (x,y), h, w
  - Cône / arc de cercle / pacman
    - centre (x,y), rayon r, start angle (start), taille arc (span)
  - Polygone

Liste de points x,y, le dernier étant relié au premier

- Dessin manueler**  
liste de points x,y, le dernier étant lié au premier
- Droite / demi-plan**
- segment(s) non fermés**  
liste de points x,y
- Zone définie par plusieurs relations**  
ex.  
 $x = a + b$   
 $y = a + c$
- Utiliser zone comme masque**  
Par rapport aux mappings...
- Charger image/matrice en tant que mapping**  
Utiliser RVB, faire mapping lumière...
  - image (r,g,b,(a))**
  - matrice (type char, int, float,...)**
  - paramètres : dim (en pixels), x,y, scale (en x et y), rotate**
- Mapping d'une adresse**
- Édition d'une zone déjà créée**
- Suppression d'une zone**

Ui:

- Paramètres d'espace**
  - Dimensions**
  - Nommage**
  - bornes & domaines de dimensions**
- Mapping espace - GUI**
  - Si dim > 2, fixer une valeur pour autres dimensions**
  - Dimension "temporelle"**  
cf. mapping de style animation
- Création de zones 'standard' par drag'n'drop**  
Depuis boutons dans une toolbar
  - Cercles**
  - Pointeurs**
  - Droites**
- Création de calculs**
  - Par valeur**
  - Par algo**
- Configuration des méthodes d'approximations utilisées**
  - Pour le rendu**

- Pour l'exécution
  - Ui d'animation
    - cf. état de l'art
  - Visualisation adaptée quand >2D
    - (en fixant des paramètres)
  - Mappings par défaut
    - Exemple : couple x, y : associé à la position d'un élément
  - Mappings adaptés aux transformations graphiques
    - Translation
    - Rotation
    - Scale
  - Si adapté : sens d'entrée / de sortie
  - Contrôle graphique à l'exécution dans plugin séparé
    - Par exemple dans un device "souris" qui expose une zone x / y
  - Template par défaut pour nouveau document
  - Affichage des mappings dans zone séparée
  - Affichage repère
  - Zoom
- Ui (édition):
- Rendre objets éditables
  - State machine adaptée
    - (ou bien les trucs de Qt suffisent ?)

Sinon la SM va dans le SpacePresenter comme il se doit.
  -
- Moteur:
- Activation / désactivation zone
  - Mappings
  - Gestion du temps comme un paramètre
    - Relatif au sous-scénario actuel ?
  - Résolution de calculs
    - Remplacement simple
    - Solving puis remplacement
    - Essayer de réfléchir à d'autres heuristiques...
  - Fonctionnement du tick
    - Relancer les calculs à chaque tick du temps ?
    - Relancer les calculs quand un message correspondant à un paramètre est rentré ?
    -

- Affichage dans device local

Example d'application:

- Avec du son

- Ajouter sources
- Position x, y, z pour chaque source
- Positionnement par OpenAL

- Arbre

/settings/volume

/settings/files

/settings/play

/settings/stop

/listener/position/{x,y,z}

/listener/orientation/{x,y,z}

/sources/add name

/sources/remove name

/sources/[name]/position

/sources/[name]/muted

/sources/[name]/file

- Metabots

- Contrôle x,y

-

## **B État de l'art.**

# État de l'art - spatial

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Pratiques</b>	<b>4</b>
2.1	Pratique appliquée à l'audio . . . . .	4
2.1.1	Composition et écriture . . . . .	5
2.1.2	Composition dans des environnements en 3D . . . . .	6
2.1.3	Acoustique virtuelle . . . . .	7
2.1.4	Restitution spatiale . . . . .	8
2.2	Pratique liée à l'interaction . . . . .	8
2.2.1	Jeu vidéo . . . . .	8
2.2.2	Interaction . . . . .	10
2.2.3	Muséographie . . . . .	11
2.2.4	Prévisualisation d'espaces . . . . .	12
2.2.5	Robots . . . . .	13
<b>3</b>	<b>Outils de manipulation et d'écriture</b>	<b>15</b>
3.1	Extraction de données . . . . .	15
3.2	Trajectoires et animation . . . . .	15
3.2.1	Trajectoires et espaces . . . . .	17
3.2.2	Spatialisation du son . . . . .	17
3.2.3	Manipulation tangible . . . . .	19
<b>4</b>	<b>Modélisation</b>	<b>20</b>
4.1	Modèles quantitatifs . . . . .	20
4.2	Modèles qualitatifs . . . . .	20
4.3	Modèles spécialisés . . . . .	22

# 1 Introduction

Ce document vise à présenter un état de l'art des pratiques impliquant la manipulation de données spatiales dans un cadre artistique, interactif et muséographique.

Nous présentons notamment les besoins des concepteurs, ainsi que les diverses solutions qui peuvent y être apportées, que ce soit par la conception de logiciels dédiés ou bien par des modélisations formelles des notions d'espace.

Le problème qui nous intéresse principalement est celui de l'écriture. Cela implique des recherches sur les applications faisant usage d'écriture spatiale, des logiciels interagissant avec des données spatiales, des modèles aptes à représenter de telles données et des paradigmes d'interfaces utilisateur permettant de les modifier et de les retravailler.

Nous définirons d'abord le périmètre auquel nous nous limitions, et étudierons les différentes utilisations de l'espace qu'il peut y avoir selon les pratiques artistiques et créatives.

En particulier, seront abordées les questions des espaces en son et musique, ainsi que dans l'écriture de l'interaction.

## 2 Pratiques

De nombreuses formes d'art sont ancrées dans le monde spatial de manière inhérente. Cependant, les réflexions sur l'espace en tant qu'objet, ainsi que sur ses limites et son influence avec le temps sont plus récentes. Dans l'art, Simona Lodi parle de la position de l'espace dans un cadre collectif[Lod14]. Par exemple, l'installation de Steve Eley est présentée : la Licorne Rose et Invisible. C'est une structure virtuelle apposée dans le monde réel et visible seulement à l'aide d'une application dédiée sur smartphone. Cela dénote une progression des pratiques artistiques qui font usage de la réalité virtuelle : la tendance est dans l'augmentation de la réalité par l'ajout d'objets qui extraient ou rendent des données (Manovich, 2005) : "Spatial Art overlays and unites several spaces into one, making artistic use of time, movement and data or information in a space defined by growth in technological interaction, i.e., a data space.".

Nous étudierons d'abord le cas de l'audio et de la spatialisation en audio, puis nous intéresserons aux questions d'espace qui apparaissent dans des applications plus centrées sur l'interactivité.

### 2.1 Pratique appliquée à l'audio

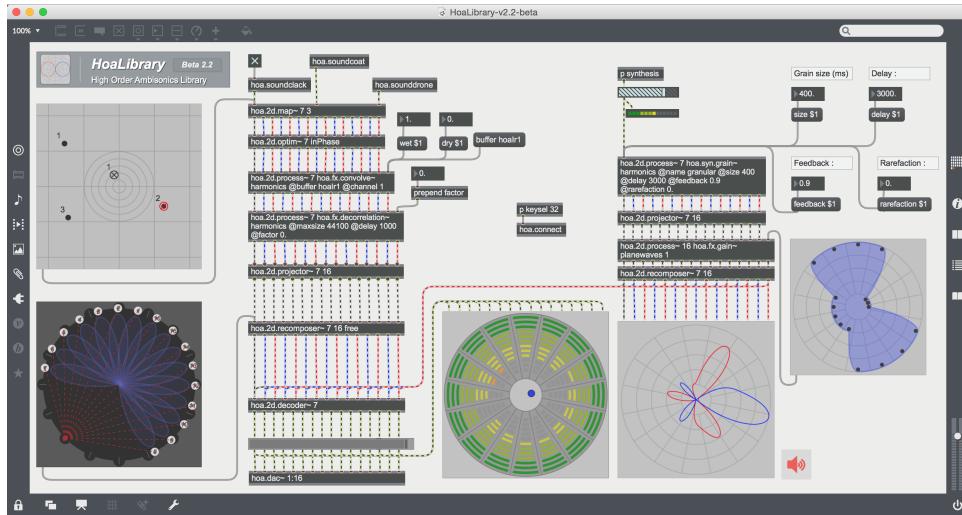


FIGURE 1 – La bibliothèque HOA.

Utilisation dans la musique, ainsi que dans des applications de réalité augmentée.

Problème principal dans cadre musical : couplage fort entre l'écriture et le moyen de restitution. Par exemple il est souvent nécessaire d'ajuster les paramètres de spatialisation quand on change l'endroit où se déroule une représentation. Ainsi la bibliothèque HOA [CGPSB13] (fig. 1) permet l'encodage et le décodage ambisonique ou binaural. Elle fournit en même temps des outils permettant de modifier et travailler la position des sons dans l'espace, et qui peuvent donc être utilisés à fin d'écriture.

Nous étudions aussi le cas de l'audio dans un cadre de systèmes interactifs : ce sont des systèmes soit très statiques comme dans un audio-guide, soit très dynamiques mais avec une boucle d'exécution simple. Par exemple, il est possible d'apposer des

couches audio dans un espace réel à l'aide de cartes pour faire des parcours[LL10] augmentés.

### 2.1.1 Composition et écriture

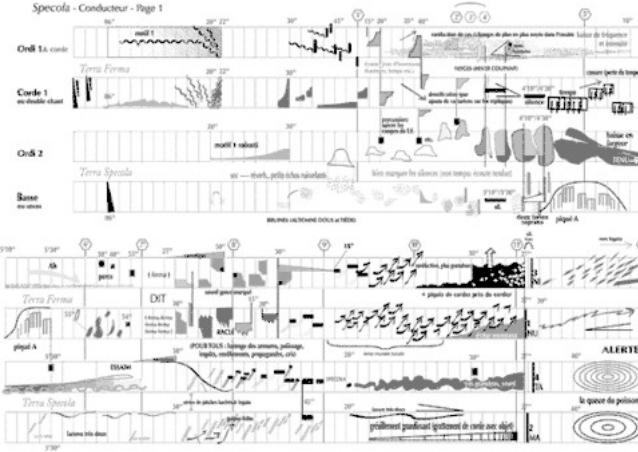


FIGURE 2 – *Specola* (LL de Mars) - une pièce acousmatique dont la partition est constituée de nombreux éléments graphiques spécialisés.

Une présentation de l'état actuel de l'écriture spatiale en musique est donnée dans [FBCG15]. Notamment, la question de la notation dans le cadre de partitions impliquant des éléments spatiaux est abordé. Ces partitions peuvent être spatiales uniquement dans leur représentation, mais peuvent aussi indiquer des manières d'interpréter dans l'espace, notamment à l'aide de symboles spécialisés [EPSZC14]. Une taxinomie des possibilités de création dans l'espace en musiques électro-acoustiques est présentée par Bertrand Merlier dans [Mer06b]. Elle est étendue dans l'ouvrage *Vocabulaire de l'espace en musique électro-acoustique*[Mer06a].

Des outils logiciels existent pour ces partitions – ils sont souvent spécialisés. Par exemple, la bibliothèque Expressive Notation Package (ENP)[KL06] permet de concevoir des partitions graphiques telles qu'en fig. 2 à l'aide d'un éditeur lui aussi graphique et d'un langage basé sur LISP.

Une des problématiques actuelles pour la représentation de l'écriture musicale est celle du geste, et de son lien avec la partition : comment notamment annoter le geste du musicien avec précision ? Et, inversement, comment à partir d'un geste créer un son correspondant ? Ces questions sont abordées dans la description de Soundstudio 4D[SSJGB04], dans le cadre d'un système de conception de trajectoires pour spatialisation à l'aide d'interactions en trois dimensions.

Une autre question est l'association entre l'aspect graphique et le résultat. Ainsi, des outils tels que HoloEdit et HoloSpat permettent de travailler avec des trajectoires, mais sont extrêmement spécialisés pour des objets audio. C'est notamment du à la nécessité de composer en ayant conscience à chaque instant des fortes contraintes techniques du moyen de restitution de l'œuvre. Il serait intéressant d'utiliser ces trajectoires pour contrôler non pas des sources sonores mais des éléments dans des espaces de paramètres quelconques.

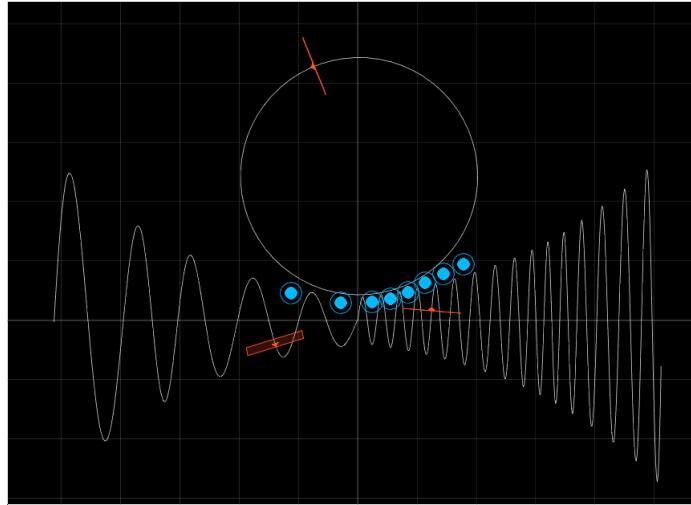


FIGURE 3 – Le logiciel IanniX.

Le logiciel IanniX[JCR12] (fig. 3) dispose aussi de nombreuses possibilités d’écriture spatiale : les partitions sont des ensembles d’éléments graphiques définis paramétriquement ou bien à l’aide d’un langage de programmation dédié, que des curseurs vont parcourir. L’information de position de chaque curseur est envoyée en OSC, ce qui permet l’intégration à d’autres logiciels.

Une méthode d’écriture de la spatialisation par contraintes est proposée par Olivier Delerue avec le système MusicScspace[DEL04]. Cela permet une approche déclarative à l’écriture de partition, en spécifiant des contraintes telles que « deux objets ne doivent jamais être à plus de deux mètres l’un de l’autre » ou bien « l’angle entre deux objets et l’auditeur doit être supérieur à 90 degrés ». Les objets peuvent être notamment des sources sonores. Une édition graphique de ces contraintes est proposée, et elles sont représentées en termes de cercles et de segments reliant les objets qu’elles contraignent.

Des données spatiales peuvent aussi être utilisées directement pour créer des mappings sonores. C’est le cas notamment de la bibliothèque Topos[NS14], qui permet de capter le mouvement de danseurs et d’en extraire des informations pouvant être utiles pour la conception de pièces de musique interactives. Une fois que le mouvement du danseur est capturé via un périphérique externe, il devient possible d’extraire des informations telles que le volume occupé par le danseur, sa vitesse, ou bien diverses mesures relatives à l’évolution de deux ou quatres points dans le temps, comme l’instabilité ou les collisions entre différentes parties du corps. Ces données peuvent ensuite être réutilisées dans Pure Data pour de la génération de musique.

Enfin, il convient de noter la richesse pour ce qui est des modes d’entrée et d’interaction. Par exemple, il existe plusieurs possibilités de composition musicale à l’aide de tables interactives comme la Reactable[KJGA06] et différentes approches dérivées qui peuvent être spécifiquement axées sur la spatialisation du son[SCV13].

### 2.1.2 Composition dans des environnements en 3D

Une approche particulière est celle de l’immersion dans un monde virtuel en trois-dimensions pour la composition de musique. Cette approche est présentée par

Michael Wozniewski dans [Woz06]. Cela permet de définir et visualiser la propagation potentielle des ondes sonores dans un espace donné. De plus, en utilisant des méthodes de tracking de mouvement et des casques de réalité virtuelle, il est possible d'ouvrir de nombreuses possibilités d'interaction manuelle avec des objets virtuels. L'implémentation est faite en PureData.

D'autres possibilités d'utilisation d'environnement 3d à des fins d'écriture musicale sont suivies par Florent Berthaut avec le DRILE[BDH10]. Notamment, l'interaction spatiale est utilisée pour manipuler une hiérarchie plus aisément que via des interactions Windows, Icons, Menus and Pointing device (WIMP) traditionnelles.

Dans les deux cas, ce sont des approches orientées vers la manipulation d'objets.

### 2.1.3 Acoustique virtuelle

Il existe plusieurs méthodes pour simuler un environnement virtuel (comme par exemple dans un jeu vidéo) sur le plan acoustique, en tenant compte de la géométrie dans laquelle peut se trouver le joueur.

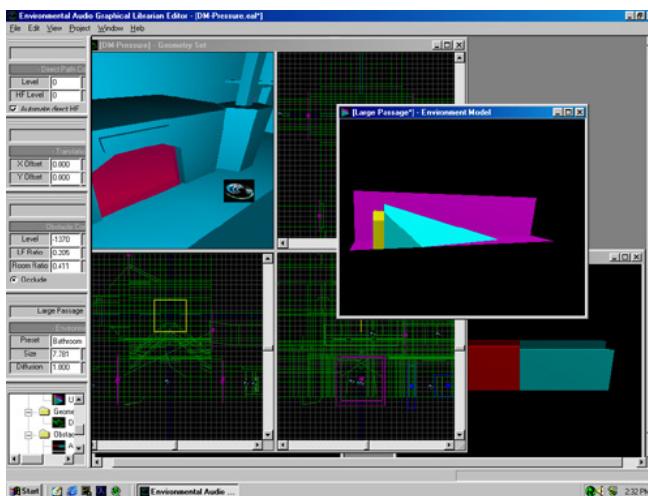


FIGURE 4 – E.A.G.L.E., l'éditeur EAX.

Les approches les plus simples utilisent simplement une atténuation linéaire entre la position du microphone virtuel, souvent appelé *listener*, et les sources sonores qu'il peut y avoir. C'était le cas notamment pour certaines implémentations d'OpenAL[Hie05]. Différents procédés, comme ceux proposés par la technologie EAX[FTJ03] (fig. 4), permettent d'ajouter une forme d'occlusion basique, ainsi que des effets de réverbération statique dans des endroits donnés pour enrichir l'impression d'espace. Des outils plus récents tels que FMOD et wWise s'intègrent notamment avec les moteurs de jeux pour pouvoir tirer parti directement des objets positionnés dans l'espace du moteur de jeu.

Cependant, il existe peu d'approches permettant d'être créatif par rapport aux possibilités qui sont offertes par l'outil informatique : notamment, les sources sont souvent ponctuelles alors que rien n'empêche de réfléchir à des sources planes ou volumétriques.

Le rendu se fait généralement à l'aide de méthodes de lancer de rayon, inspirées des procédés utilisés en images de synthèse [FCEPSW98 ; TG98]. Un procédé courant

est maintenant d'utiliser les Graphics Processing Unit (GPU) pour effectuer des calculs lourds mais facilement parallélisables [Rod14 ; Che14 ; TCMLSM12]. Il est aussi possible de précalculer une fonction de convolution correspondant aux environnements virtuels que l'on veut parcourir, cependant cela demande des ressources en mémoire très importantes [RS14].

#### 2.1.4 Restitution spatiale

Il existe de nombreuses méthodes pour effectuer un rendu de son spatial. Certaines s'intéressent au rendu d'une "scène" spatiale donnée le mieux possible à l'aide d'un certain nombre de hauts-parleurs, tandis que d'autres visent la simulation d'une acoustique virtuelle comme nous l'avons vu précédemment. Comme pour HOA, les outils de création sont souvent fortement associés au système de rendu. La bibliothèque 3Dj [Per15] pour Supercollider vise à séparer cela à l'aide d'une architecture modulaire séparant le mapping d'interface utilisateur, la gestion de scène spatiale, les comportements, les modèles physiques, l'export et le rendu via méthodes ambisoniques, HRTF [NMSH03], Vector Base Amplitude Panning (VBAP), Distance-Based Amplitude Panning (DBAP) et Wave-Field Synthesis (WFS). La communication entre la gestion de la scène et le rendu se fait via le format standardisé SpatDIF [PLS13] par OSC.

Une autre proposition d'architecture modulaire, par strates, est proposée par Nils Peters dans [PLSBBP09].

Les différentes méthodes de spatialisation citées sont comparées via la capacité des auditeurs à situer la présence d'une source sonore dans l'espace dans [BBFK07], et la précision de leur localisation.

Une méthodologie plus orientée sur la notion d'objets sonores que sur le mixage de canaux est présentée dans [Kim14]. Elle repousse la projection en canaux des objets au tout dernier stage du rendu, juste avant l'envoi aux hauts-parleurs. Cela permet de s'adapter plus facilement à des configurations de haut-parleur varié, néanmoins cette métaphore est plus adaptée au monde du jeu vidéo et de la réalité mixte, qu'à la restitution musicale ou le travail par pistes est une pratique dominante.

L'intérêt de telles restitutions n'est pas seulement artistique : l'audio spatialisé peut être utilisé pour combler les déficiences visuelles, par exemple via un processus de sonification [TL14]. Cela implique notamment une phase de reconnaissance d'objets, puis une phase de mapping d'un espace visuel défini hiérarchiquement vers un espace de sons qui permettra à la personne malvoyante de reconnaître des objets inertes via le son qui leur sera associé.

## 2.2 Pratique liée à l'interaction

### 2.2.1 Jeu vidéo

La notion d'espace (et plus généralement, d'espace-temps) intervient à plusieurs échelles lors de la création de jeux vidéos. Elle est présente notamment à l'échelle du niveau de jeu, et décrit directement les interactions entre le personnage et son environnement, comme ce qu'il se passe lors d'une rencontre avec un ennemi ou un passage sur un objet bonus. Mais une notion plus large de narration spatio-temporelle va parfois sous-tendre le déroulement du jeu. Liliana Salazar décrit dans

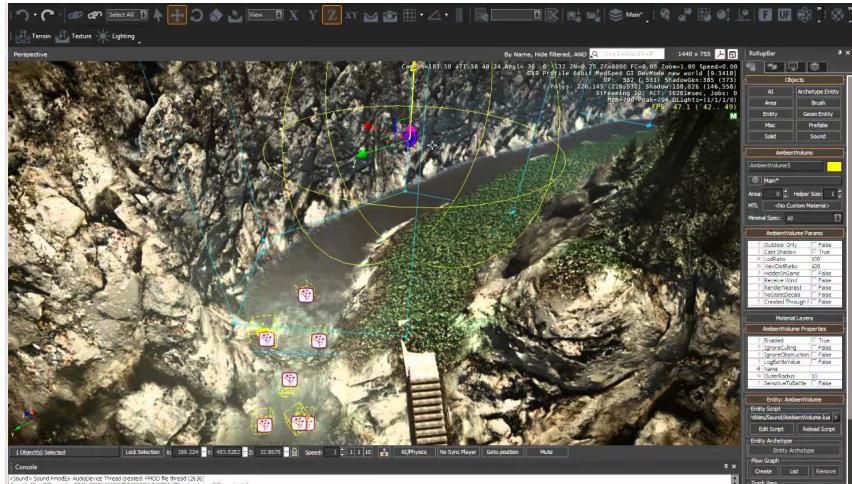


FIGURE 5 – Positionnement de sources sonores dans le CryEngine

sa thèse [Sal04] différents modèles qui peuvent servir à décrire formellement un jeu vidéo ; notamment des modèles basés sur les réseaux de Petri pour décrire les niveaux d'un jeu. Cela permet d'appréhender la complexité de telles œuvres : les jeux vidéos mêlent logique, temporalité, et spatialité.

Cécile le Prado s'intéresse plus particulièrement au lien entre les espaces visuels et sonores dans les jeux vidéos[Le 13] par le biais de la cartographie.

Enfin, les moteurs de jeux généralistes tels que Unreal Engine, Unity, Unigine, Cry Engine, ont généralement des outils rudimentaires pour le positionnement du son (fig. 5). Mais du fait de leur extensibilité, il existe une grande quantité d'outils implémentant divers algorithmes de spatialisation, comme par exemple 3Dception<sup>1</sup> (fig. 6) ou IOSONO Proximity<sup>2</sup>. Cependant, il convient de noter que la partie qui est la plus travaillée durant la conception d'un jeu est généralement la partie graphique, et que le son spatialisé, malgré tout ce qu'il peut apporter, est traité comme un citoyen de seconde zone.

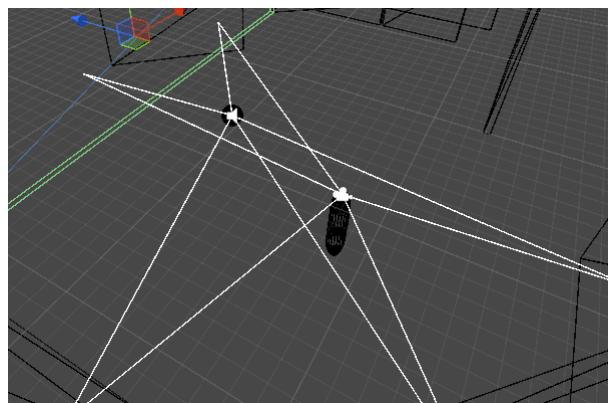


FIGURE 6 – Réflexion du son dans 3Dception.

De plus, dans de nombreux cas il n'y a que peu de travail sonore nécessaire :

1. [http://www.twobigears.com/3dception\\_unity.php](http://www.twobigears.com/3dception_unity.php)
2. <http://www.iosono-sound.com/game-audio>

l'idée est de recréer un environnement réaliste, ce qui ne permet pas une marge de créativité énorme, mais demande plutôt des algorithmes précis.

### 2.2.2 Interaction

La majorité des outils que nous avons vu jusqu'à présent sont des outils de création qui fonctionnent peu ou prou avec le paradigme d'interface WIMP, modulo des spécificités pour la création de jeux ou d'univers en 3D par exemple. Cette partie présente plus en avant les possibilités d'interaction qu'il peut exister dans le domaine spatial.

Nous distinguons deux aspects : la conception de contenu spatial est par essence une activité interactive que nous désirons étudier. Mais dans certains cas, on désire concevoir des applications offrant une part d'interactivité à l'utilisateur final, comme cela est le cas dans les bornes interactives par exemple. L'interaction étant souvent gestuelle, un modèle spécifique d'espace géré en temps réel mais conçu à l'avance devient alors nécessaire. Ces deux aspects sont donc liés : une des thématiques qui nous intéresse est la conception d'outils de conceptions, notamment d'interaction. Il serait possible de parler de métaconception.

Un état de l'art très complet pour l'interaction homme-machine dans un cadre de réalité mixte est présenté dans la thèse de René Chalon [Cha04]. Outre une description des différents périphériques d'entrée-sortie disponibles à l'époque, sont présentés différents modèles adaptés à la gestion de l'interaction dans la conception d'une application en réalité mixte. Notamment, le modèle IRVO est introduit, et permet la séparation entre les objets réels, manipulés par l'utilisateur dans le monde physique, et ceux, virtuels qui sont uniquement présents dans la machine et représentés par projection. Une grande complexité réside dans l'interaction entre les deux espaces que sont le monde réel et virtuel, et le mapping d'informations de l'un à l'autre. Ces questions ont notamment été étudiées par Michel Beaudouin-Lafont avec la théorie de l'interaction instrumentale.

Un second état de l'art, plus récent et axé spécifiquement sur l'interaction qu'il peut y avoir avec les environnements 3D est présenté dans [JH15]. Les méthodes d'interaction sont séparées entre celles qui ciblent la navigation dans un espace 3D, celles qui visent la sélection et manipulation de données, et celles qui permettent le contrôle des systèmes intermédiaires dans un environnement axé sur la 3D.

Notamment, il existe des interfaces extrêmement spécifiques, qui peuvent souvent être utilisées à but artistique. Par exemple, le système BlowBrush[SL14] utilise un moulin à vent interactif pour peindre des feuilles sur un canevas virtuel.

Enfin, le développement des interfaces imaginaires[GBB10], qui superposent des informations virtuelles au monde réel sans pour autant avoir de forme visible, ouvre de nouvelles possibilités en terme de conception spatiale, car il devient possible de tricher quant à la localisation réelle des objets : dans le jeu de basket-ball imaginaire présenté, la position de la balle est par exemple régie par un système probabiliste et l'acquisition d'information par les joueurs se fait via le son.

### 2.2.3 Muséographie

Dans le cadre d'applications muséographiques, les informations spatiales sont au premier plan. Notamment, les points clefs sont :

- La conception d'interfaces utilisateur accessibles et adaptées à différentes classes d'âge. Michael Despina présente plusieurs faits intéressants, via une comparaison d'installations interactives dans [MPCZHC10]. Par exemple, pour une installation donnée, ce n'est pas parce qu'il y a un grand nombre de degrés de liberté en entrée ni que le contenu est en 3 dimensions que des enfants prennent du plaisir et ont envie de revenir participer à l'installation. C'est plutôt le type d'activité proposée qui va déterminer la qualité de l'expérience, peu importe le médium. De plus, l'auteur note une certaine réticence des jeunes enfants à approcher d'eux-mêmes les installations basées sur un écran par opposition aux formats en table interactive, notamment due à de potentielles restrictions sur l'utilisation d'ordinateur à domicile.
- Des visualisations et univers graphiques et sonores spécifiques à chaque projet.
- Des modes d'interaction variés. Notamment, la place de l'interaction multi-touch, désormais très utilisée, est étudiée dans [KNL11].



FIGURE 7 – Medialon KIOSK, un outil de conception de bornes interactives.

Le parc logiciel repose généralement sur des outils graphiques comme Adobe Flash, des bibliothèques spécialisées comme Processing ou OpenFrameworks, et des outils de conception génériques tels que Medialon (fig. 7).

Dans le cadre de parcours muséographiques et d'enrichissement du patrimoine culturel, il existe différentes approches qui visent à augmenter les œuvres et présentations. Par exemple, il est possible de sonifier des tableaux par rapport à différentes méthodes de segmentation, telles que des méthodes simples par clef de couleur. C'est le cas pour le logiciel SUM[AK12]. Ce logiciel permet d'écrire de véritables partitions basées sur des tableaux, avec un système d'évolution dans le temps et en fonction du geste, qui permet de redécouvrir des tableaux sous un angle sonore.

Une autre approche pour l'enrichissement des visites de musées est présentée dans [Azo14]. L'approche vise à concevoir des paysages sonores à l'échelle d'une pièce ou même du musée entier, qui sont en lien avec le thème et se raffinent lorsqu'une œuvre en particulier est contemplée par les visiteurs. Un modèle (fig. 8) de visite de musée basé sur XML, permettant notamment de définir le degré de libertés accordé

au participant est introduit. Il sépare, dans le cas de la scène sonore, les couches physiques et virtuelles qui ont déjà été abordées, ainsi que la couche sémantique.

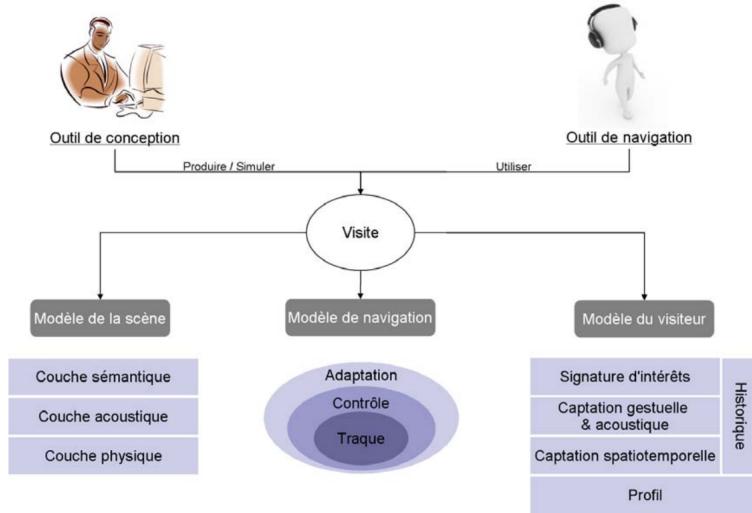


FIGURE 8 – Modèle de structuration de visite de musée de Fatima Kaghat. Les trois modèles ont besoin d’outils appropriés de conception et de réaction adaptés à des données spatiales.

#### 2.2.4 Prévisualisation d’espaces

Lors d’installations ou de spectacles complexes, une des grandes problématiques est celle de la prévisualisation, puis de la répétition. Une possibilité est de concevoir des outils de storyboard qui permettent de définir comment va se dérouler une pièce.

Cela pose de nombreuses questions quand au rôle de ces outils dans un cadre interactif. Par exemple, comment gérer le storyboard de spectacles interactifs avec un certain nombre de libertés laissées aux acteurs ? Les problèmes les plus importants sont ceux de la synchronisation et de la remise dans un état donné.

Le framework ANSWER, présenté dans [JWJBF10], vise à fournir des outils pour le storyboard de films, en partant du langage Director Notation permettant de spécifier le déroulement d’un film sur le plan conceptuel. Le problème est notamment celui d’outil de spécification et conception permettant des esquisses de plans de caméra, de positions de décor et d’éclairage, etc. L’approche utilisée est basée sur le système déclaratif orienté web X3D qui sera présenté en dernière partie de ce document. D’autres langages, tels que TVML (en figure 9), sont aussi utilisés pour différents stades de la pipeline de production de contenu télévisuel et cinématographique.

Un autre exemple de logiciel permettant l’écriture et la visualisation de scénographies est StageViz[LH13], développé en Corée du Sud. L’intérêt de ce logiciel est son interopérabilité avec divers logiciels d’écriture, ce qui permet d’avoir un rendu graphique proche de ce que sera l’exécution réelle du spectacle à condition de fournir des modèles 3d adaptés et de fournir une brique d’intégration si un protocole spécifique est nécessaire. L’écriture se fait par successions d’états envoyés via des protocoles dédiés comme DMX.

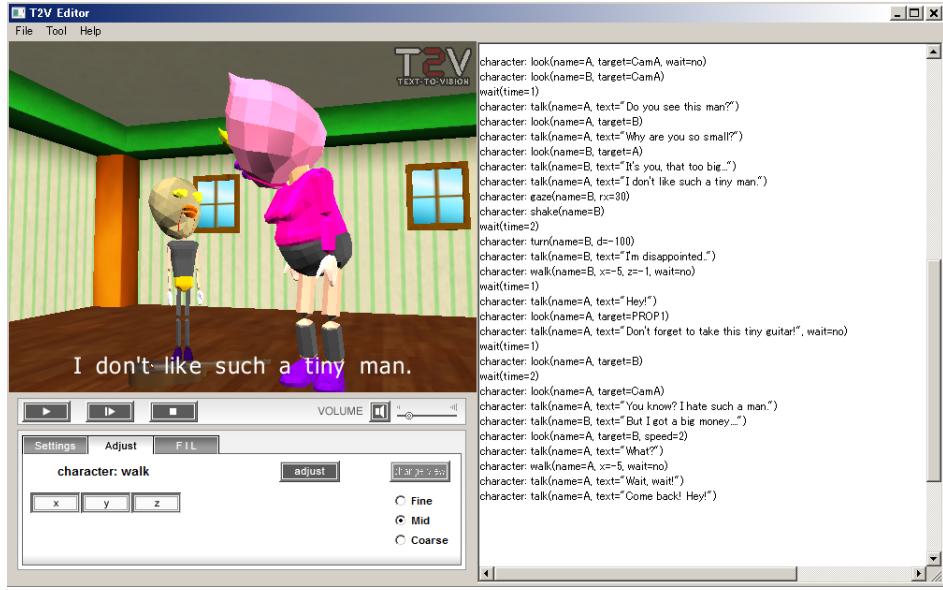


FIGURE 9 – Un exemple d’application de conception TVML, avec une partie prévisualisation et une partie langage. Ici, le langage est extrêmement spécialisé.

### 2.2.5 Robots

L’écriture spatiale prend tout son sens lorsque l’on désire travailler avec des robots. De plus en plus d’artistes tentent de lancer des projets impliquant des robots musiciens, danseurs, ... et il est nécessaire de concevoir des outils adaptés. Notamment, les robots ont d’une part de fortes contraintes physiques dont il est nécessaire de tenir compte lors de l’écriture, et d’autre part, sont faillibles ; il est nécessaire de concevoir des outils permettant d’adapter le spectacle lors d’une chute par exemple.

Un premier exemple est donné dans [LKH13]. La question posée est surtout celle de la conception de robots adaptés à des spectacles : un outil graphique est présenté, qui permet de concevoir facilement des robots par rapport à des modèles réels en combinant des moteurs virtuels en 3D. De plus, le robot virtuel réponds aux mêmes signaux que le robot réel. Il est alors possible de simuler la chorégraphie robotique sans avoir de problèmes de casse en cas de collision.

Cet outil est développé plus en avant dans [LKKH14] : un modèle permettant de calculer le couple appliqué par les moteurs du robot virtuel est rajouté au système, ce qui permet d’éviter d’autres problèmes au moment de l’écriture. Pour cette étude, une comparaison probante a été menée entre deux robots réels et leurs avatars virtuels, en utilisant un système de motion capture.

Il est aussi possible d’utiliser des méthodes de mapping pour laisser plus de liberté au robot. Ainsi, dans [SYKK13] un système de reconnaissance de musique générant des mouvements de danse en rythme est présenté. Cela part du constat que la plupart des chorégraphies pour robot ont actuellement besoin d’être programmées à la main, ce qui est particulièrement fastidieux. La chorégraphie est ici annotée à l’aide de Labanotation (fig. 10), qui est une notation pour la danse, et reposant sur 12 mouvements basiques du corps humain. La forme des symboles rappelle la position du corps et des articulations.

Enfin, les dernières avancées technologiques permettent d’utiliser des drones en

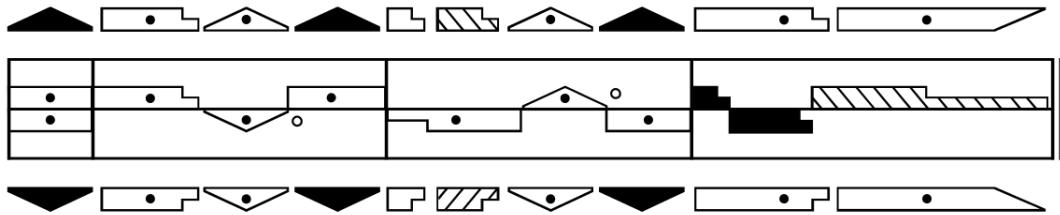


FIGURE 10 – Exemple de Labanotation pour la danse. Il est possible de spécifier la direction, l'intensité, la durée et la partie du corps correspondant à un mouvement.

plus de robots traditionnels. Une utilisation intéressante est celle des Spaxels [HGHLB12] : chaque drone devient une sorte de pixel spatial, qui porte une couleur et peut se déplacer comme il le désire dans un volume donné (fig. 11). Un des problèmes majeurs lors de l'écriture d'une telle chorégraphie sera la gestion de l'inertie et des collisions : là encore, un outil de prévisualisation prenant en compte les lois physiques et les conditions que peuvent rencontrer les drones sera nécessaire.



FIGURE 11 – Des spaxels réalisent une théière dans l'espace à l'aide de trainées lumineuses.

### 3 Outils de manipulation et d'écriture

Jacob Beal présente dans [BV15] les possibilités de gestion de l'espace-temps dans des applications à l'aide d'appareils connectés. Ces objets, comme les smartphones et smartwatches, étaient auparavant simplement connectés à un réseau, mais ont maintenant des informations de localisation et position, ainsi que de nombreux autres capteurs très précises avec lesquelles il est possible de concevoir des applications opérant sur de larges échelles géographiques. L'approche utilisée est de traiter l'ensemble des appareils connectés comme un champ, et de programmer des applications en utilisant les lois physiques qui régissent ces champs.

#### 3.1 Extraction de données

Une possibilité couramment employée pour l'écriture est celle du mapping, c'est-à-dire que des données sont extraites d'un premier médium, transformées, puis appliquées à un second médium. Cela peut être, par exemple, la transformation d'un mouvement en image ou d'un son en un autre son.

À cet effet, il est utile d'avoir des métriques que l'on peut utiliser comme source de mapping. À titre d'exemple, Congcong Li présente dans [LC09] 26 différentes métriques qui peuvent être extraites de tableaux et qui permettent de mesurer les qualités esthétiques, et par la suite de classifier les tableaux par style. La notion d'esthétique est empirique : dans ce travail, elle est basée sur une étude avec 42 participants et 100 tableaux ; les participants devaient dire pour chaque tableau s'il était de bonne ou mauvaise qualité puis répondre à un questionnaire détaillé sur les facteurs de qualité perçus. Par la suite, cela a permis de chercher des métriques pouvant influencer ces aspects plus détaillés. Des exemples de métriques extraites sont la clarté moyenne logarithmique, ou bien la forme des segments d'un tableau après segmentation algorithmique.

Il serait par la suite intéressant d'appliquer ces métriques en temps réel à des flux vidéos ; cela fournirait un espace de paramètres importants avec lequel travailler.

#### 3.2 Trajectoires et animation

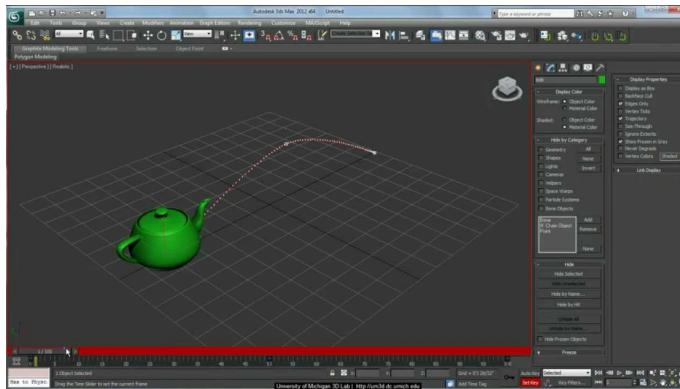
Une des problématiques les plus importantes est celle de l'écriture des trajectoires. Pour ce faire, il existe de nombreuses approches que nous séparons en deux parties : celles axées sur une entrée directe qui est ensuite analysée, par exemple avec un stylo électronique, et d'autres basées sur des méthodes mathématiques comme des équations paramétriques.

Rhonda<sup>3</sup> rentre dans la première approche. C'est un logiciel de dessin en trois dimensions à partir d'une entrée en deux dimensions et d'une source de rotation comme un gyroscope ou une souris adaptée. Il est utilisé dans Sonic Wire Sculptor et Ink Space, détaillé dans [Ras13]. Une telle méthode d'entrée serait efficace pour définir des trajectoires spatiales car elle demande une interaction très simple.

C'est en opposition avec les logiciels généralistes d'animation graphique, vectorielle ou vidéo. Les techniques les plus utilisées dans ce cas sont celles des keyframes et du rotoscoping (fig. 12).

---

3. <http://rhondaforever.com/>



(a) Keyframes



(b) Rotoscoping

FIGURE 12 – Modes d’animation standards : keyframes et rotoscoping

La problématique principale en animation est de lier l'espace et le temps. Les méthodes par keyframes donnent priorité au temps, tandis que les méthodes par rotoscopie sont centrées sur des modifications d'espaces. De plus, dans les outils existants, le contrôle du flot du temps est généralement séparé de la création des données spatio-temporelles (les trajectoires).

Une solution est proposée dans [SCBS13], en utilisant un crayon pour contrôler le temps via les trajectoires lors de l'animation.

Une autre possibilité, implémentée dans Kitty[KCGF14] repose sur la création de textures kinétiques : les textures, qui généralement ne sont que des informations de couleur, contiennent ici une information de déplacement. Plusieurs types de textures kinétiques sont présentés : des textures émettrices qui génèrent des objets, ainsi que des textures oscillantes qui animent un objet donné. Un graphe relationnel est ensuite utilisé pour définir les interactions qu'il peut y avoir entre les différentes entités de l'animation. Par exemple, cela permet de faire rebondir une balle lorsqu'elle « tombe » sur une table. Il est ainsi possible d'avoir des animations dynamiques et interactives qui peuvent répondre à une stimulation extérieur, après création (fig. 13).

Une description interactive de trajectoires représentant des mouvements contraints par les lois physiques habituelles en 2D est explorée avec PhysInk[SD13]. La cible est notamment l'enseignement de la physique et de la mécanique plane et le logiciel est sensé fonctionner comme un tableau blanc, mais qui possède une gestion

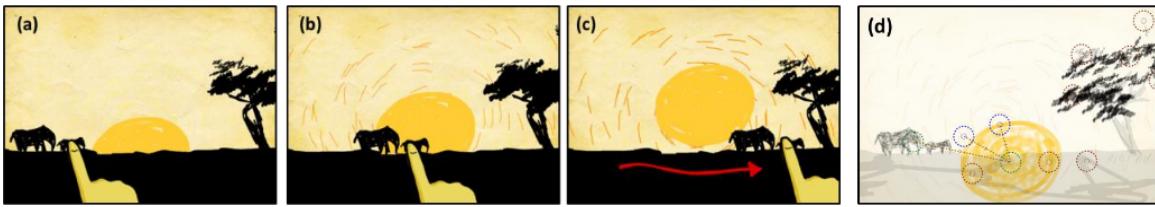


FIGURE 13 – Un exemple d’animation dynamique avec Kitty. Le soleil se lève lorsque l’éléphant est déplacé.

des collisions réaliste. Dans ce système, l’utilisateur décrit un comportement en le dessinant. Les modifications faites sont gardées en mémoire : cela devrait permettre, à terme, d’effectuer des comparaisons entre plusieurs mouvements décrits dans un même document.

Enfin, la question de la représentation d’une animation se pose. En effet, il est souvent important pour l’auteur d’avoir non seulement l’état actuel mais aussi l’état passé et futur d’un élément lorsqu’il se positionne à un point  $t$  dans le temps. Ce n’est pas problématique pour une trajectoire qui ne serait que fonction du temps, et donc qui peut être représentée comme une ligne dans un espace — même s’il existe un problème de représentation évident d’un espace 3D sur un écran 2D, pour lequel des pistes sont apportées dans [CMZHB99]. En revanche, l’animation d’éléments graphiques ou bien dans plus d’une dimension, comme par exemple une animation au cours de laquelle un objet subirait un morphing, pose des problèmes de représentation. Une solution adaptée à la représentation des animations suite à capture de mouvement est présentée dans [CTGH13]. Les logiciels d’animation standard se contentent généralement d’afficher les  $n$  images précédentes et suivantes avec une opacité décroissante. Ces problèmes sont encore accrus dans le cas où l’on aimerait prendre en compte l’interactivité : dans ce cas, il n’y a plus qu’une seule ligne de temps, mais un ensemble d’évolutions possibles pour les objets.

### 3.2.1 Trajectoires et espaces

- État de l’art sur l’édition de courbes 3D en séparant en deux axes : le dispositif physique (entrée - sortie) et le langage d’interaction (création / édition de courbe, et manipulation de la caméra). [JBLFC14]

### 3.2.2 Spatialisation du son

Nous nous intéressons maintenant au cas précis de l’écriture de trajectoires pour le son. Ces trajectoires décrivent le déplacement d’objets sonores invisibles mais dont on est sensé pouvoir donner la position lorsqu’on écoute la pièce.

Une étude menée avec plusieurs compositeurs a permis de cibler certains des besoins qui sont exprimés lors de la pratique d’écriture de musique à l’aide de contraintes spatiales. Cela est fait dans ce cas précis par le biais du logiciel Trajectories[FGB15] qui fonctionne sur dispositifs mobiles multitouch, et est implémenté avec des technologies web.

Un modèle plus complet d’espace en trois dimensions est fourni par COSM[WS11]. Implémenté dans Max/MSP, il offre une grande richesse d’écriture. En plus de lieux

et de trajectoires, il est possible d'écrire l'interaction dans une certaine mesure, ainsi que la communication entre différents agents. Une nouveauté est la possibilité de travailler avec des champs définis mathématiquement. Ces champs peuvent varier dans le temps et être sonifiés par la suite.

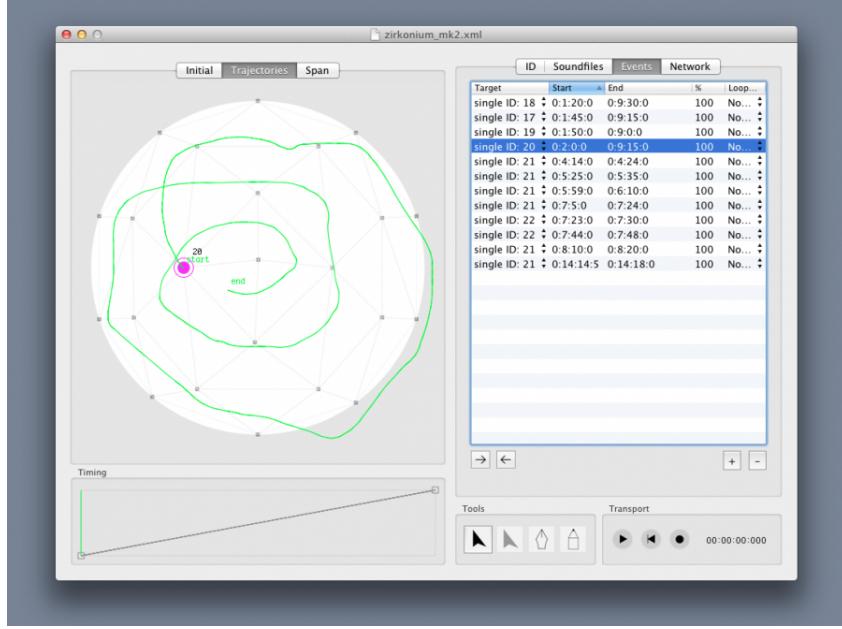


FIGURE 14 – Éditeur de trajectoires du Zirkonium Mk. 2.

Le Zirkonium, en fig. 14 est un ensemble d'applications pour Apple OS X à l'origine conçues pour spatialiser le son du Klangdom, un lieu otpimisé pour la diffusion multi-canale du son. C'est un système logiciel basé sur une architecture client-serveur, et pourvur d'un éditeur de trajectoires spatiales pour le son qui permet notamment d'utiliser des courbes de Bézier pour l'écriture[WBDO14]. Les briques logicielles communiquent par OSC ce qui permet une certaine souplesse.

Un autre système de spatialisation, cette fois basé sur la réalité augmentée et utilisant la méthode WFS est présenté dans [MLV05]. Notamment, une liste d'interactions possibles est présentée, ainsi que leur résultat dans le système. Il est notamment possible de disposer spatialement les sources sonores, d'ajuster des effets audio, et de simuler l'acoustique d'autres pièces en interagissant avec des objets dont la forme est concue pour avoir un lien avec les son auxquels ils se rapportent.

OpenMusic, avant d'être un système permettant spatialisation, est un outil très puissant de composition informatique. Basé sur la forme graphique du *patcher*, à l'image de Max/MSP et PureData, il agit de manière entièrement hors-ligne. Le paradigme de programmation sous-jacent est fonctionnel (cela permet une conception simple du flot de données) et orienté objet. Depuis quelques années, OpenMusic est doté d'outils pour la composition de musique avec des composantes spatiales[Bre12]. Le modèle utilisé est une matrice joignant les sources sonores à des paramètres de spatialisation, qui vont évoluer dans le temps. Les trajectoires sont définies en coordonnées cartésiennes, et peuvent être dotées d'un tag temporel qui permet de faire le lien avec l'écoulement du temps dans le morceau. Cela se rapport à la notion de keyframe vue précédemment. Les données spatiales ont la particularité de pouvoir

être utilisées aussi lors de la synthèse des sons, et non pas seulement comme un filtre qui serait appliqué après génération comme c'est le cas dans la majorité des autres approches.

### 3.2.3 Manipulation tangible

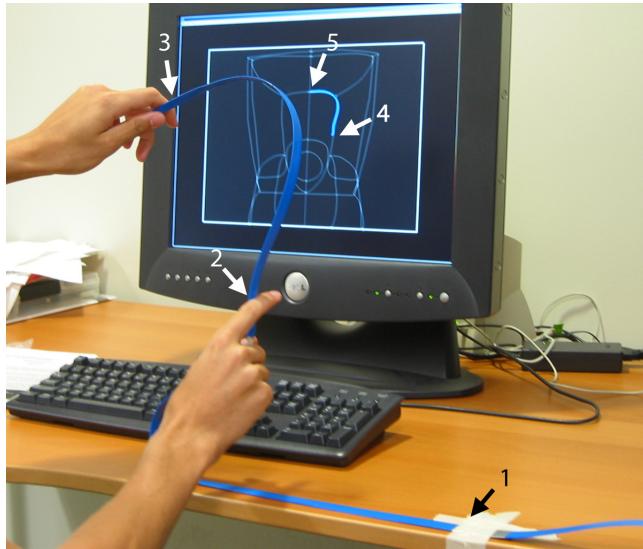


FIGURE 15 – Utilisation de ShapeTape.

Outre les interactions WIMP, prépondérantes, et les modèles basés sur la réalité virtuelle ou augmentée, il existe aussi des interfaces tangibles. Notamment, un périphérique a été spécialement conçu pour le design de courbes : ShapeTape[GBS03], un ruban déformable qui permet d'obtenir informatiquement sa torsion en 32 points pour reconstruire la courbe de manière virtuelle en 3D (fig. 16). Un vocabulaire de mouvements possibles (torsions, étirements, poussées, ...) est défini.

## 4 Modélisation

Il existe trois grandes familles de modèles utilisés pour la représentation de données spatiales :

- Les modèles quantitatifs : ce sont simplement les modèles basés sur des coordonnées cartésiennes.
- Les modèles qualitatifs : ils s'intéressent aux relations entre objets dans l'espace. Notamment, des formes de logique spécialisées sur l'espace existent et seront présentées plus en avant. Un ouvrage de référence sur ce sujet est le *Handbook of Knowledge Representation*[PLV08]. Ces outils sont souvent adaptés aux problèmes d'intelligence artificielle, et de reconnaissance d'image.
- Les modèles spécialisés : souvent issus du génie logiciel, ils visent à répondre à un problème précis, comme la localisation géographique à une certaine précision ou des systèmes en réalité virtuelle et augmentée.

Enfin, il existe des modèles qui ciblent principalement la définition de l'interaction. Notamment, dans [BF93] est présentée une liste de notions qui servent à définir les comportements possibles dans un univers en réalité virtuelle. Parmis les notions présentées, nous retiendrons celle d'aura qui permet de définir un espace autour des objets qui les rend sensibles à l'interaction. Par la suite, les notions de focus et de nimbus sont introduites, qui permettent de rendre compte de la consicence que deux objets vont avoir l'un de l'autre, ce qui permet des interactions riches.

### 4.1 Modèles quantitatifs

Le cas le plus courant que nous avons vu jusqu'à présent est celui de la trajectoire d'un objet dans l'espace par rapport au temps. L'outil évident est l'équation paramétrique : souvent, les fonctions splines sont utilisées pour interpoler des trajectoires entre des keyframes. Mais certains outils, tels que IanniX, permettent de définir textuellement via le langage mathématique n'importe quelle équation qui est fonction du temps.

Cependant, il peut être intéressant de considérer des cas plus généraux que ceux des simples trajectoires. Notamment, pour la conception d'espaces, il y a deux approches : les approches vectorielles et géométriques, qui sont souvent basées sur un petit nombre de primitives composées et transformées, et l'équivalent volumique de la piexlisation, la voxelisation.

Cela consiste à diviser l'espace en voxels[KCY93], qui est le diminutif de Volumetric Pixel. Il existe de nombreuses manières d'optimiser un espace voxelisé.

### 4.2 Modèles qualitatifs

Les modèles qualitatifs visent à représenter l'espace de manière purement logique. Plusieurs ont pour base, ou au moins pour source d'inspiration l'algèbre d'intervalles définie par Allen[All84]. L'avantage est une facilité pour les raisonnements formels sur l'espace. Cependant, l'utilité de ces modèles se trouve plus dans l'analyse de données existantes, dans un contexte sémantique par exemple, que dans la création de nouveaux contenus qui seront spécifiés plus précisément par des auteurs. Les

modèles principaux sont décrits dans [CCLWOY15]. Sont notamment couvertes les notions de relations entre objets, de direction, de distance, de mouvement et de forme.

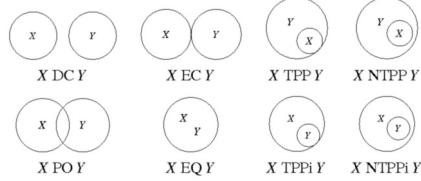


FIGURE 16 – Les relations de RCC-8. Il existe un second modèle, RCC-5, qui ne tient pas compte des différents cas de tangence.

Un modèle courant, Region Connection Calculus-8 (RCC-8) est basé sur la notion de connection topologique entre objets[Sch96 ; DM14]. L'avantage de tels modèles est notamment leur indépendance au nombre de dimensions. Les applications sont courantes dans le domaine des services d'information géographique[BW14]. Des extensions existent notamment pour spécifier l'interaction avec une dimension temporelle[Haz12], ainsi qu'avec non plus des certitudes mais des probabilités de connection[GA12].

Plus généralement, la question d'une ontologie pour les modèles spatiaux se pose. Une approche est offerte dans [Gre03]. Cet article présente la construction d'une ontologie du raisonnement spatio-temporel basée sur la notion de réalité. Cela est fait en séparant entre évènements *continuants* et *occurents*. Ces composants de l'ontologie générale présentée servent de point de vue sur la réalité et peuvent ou non interagir avec les autres. La persistence des continuants se fait par l'existence de parties temporelles successives ; le temps est une dimension de spécification des occurents mais pas des continuants.

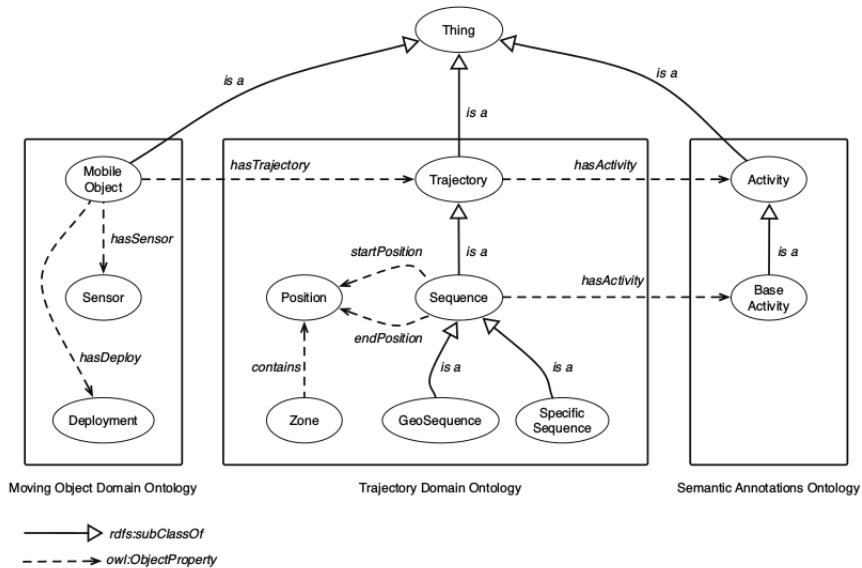


FIGURE 17 – Extrait d'une ontologie des trajectoires spatio-temporelles.

Une autre ontologie, visant à être adaptée aux trajectoires dans l'espace(fig.17), est présenté par Wafa Mefteh dans [Mef13]. Un des objectifs est notamment de

modéliser et traiter les trajectoires issus de capteurs d'appareils mobiles. Le langage OWL est utilisé pour la modélisation et de multiples autres ontologies adaptées à diverses applications spatiales, notamment la géographie, sont présentées dans cette thèse. Cela peut donc nous permettre d'étudier des modèles qui seraient plus spécifiquement adaptés aux problèmes artistiques évoqués dans ce document.

### 4.3 Modèles spécialisés

Ici, nous nous intéressons plus à des modèles qui ont été spécialisés pour réaliser une tache avec élégance. C'est donc sans surprise que nous retrouvons plusieurs langages spécifiques à un domaine (DSL) développés en Haskell. Le premier est à l'origine d'un courant de programmation : décrit par Paul Hudak, la programmation fonctionnelle réactive peut être utilisée pour contrôler des systèmes spatiaux en temps réel. Un exemple de contrôle appliqué aux robots est donné dans [HCNP03].

Un second cas propose une modélisation fonctionnelle des animations[MG11]. Ce langage part du principe que toute animation possède un début, un milieu et une fin, propose une construction adaptée à une telle conception via le langage *Active*, et présente les différents avantages que cela peut avoir en terme d'écriture. Notamment, la combinaison d'animation est aussi simple à exprimer que la composition de fonctions. Puis, un second langage adapté au rendu, *ChalkBoard*, est présenté.

Un autre cas de modèle de données, adapté à des informations spatiales, est X3D[JRSJBS13]. C'est un langage basé sur XML, incorporable dans une page web pour afficher des informations 3D de manière fluide. Par opposition à WebGL, qui correspond à l'interface de programmation impérative d'OpenGL, X3D est purement déclaratif. Il est analogue au standard SVG.

Enfin, les différents outils de spatialisation audio qui ont été présentés ont, dans un objectif de collaboration et de facilité, oeuvré vers la conception d'un standard pour les informations de spatialisation : Spatial Data Interchange Format (souvent abrégé SpatDIF ou SDIF dans la littérature)[KPGTB08 ; PLS13]. SpatOSC, Une bibliothèque de spatialisation en C++, a été introduite par Wozniewski dans [WQS12]. Elle permet de convertir entre différents formats de données spatiales pour l'audio, .

## Références

- [AK12] Sara ADHITYA et Mika KUUSKANKARE. Composing graphic scores and sonifying visual music with the SUM tool. In *Proceedings of the 9th sound and music computing conference (SMC 2012). copenhagen, denmark*, 2012, pages 171–176. URL : <http://www.smcnetwork.org/system/files/smc2012-181.pdf> (visité le 12/04/2015).
- [All84] James F ALLEN. Towards a general theory of action and time. *Artificial intelligence*, 23(2) :123–154, 1984.
- [And12] Tahiry ANDRIAMAROZAKANIAINA. Du texte à la génération d'environnements virtuels 3d : application à la scénographie théâtrale. Thèse de doctorat. Université Toulouse le Mirail-Toulouse II, 2012.
- [Azo14] Fatima-Zahra AZOUGH. Modèle et expériences pour la visite des musées en réalité augmentée sonore. Thèse de doctorat. Paris, CNAM, 2014. URL : <http://www.theses.fr/2014CNAM0926> (visité le 02/11/2015).
- [BBFK07] Enda BATES, Frank BOLAND, Dermot FURLONG et Gavin KEARNEY. A comparative study of the performance of spatialization techniques for a distributed audience in a concert hall environment. In *Audio engineering society conference : 31st international conference : new directions in high resolution audio*. 00013. Audio Engineering Society, 2007. URL : <http://www.aes.org/e-lib/browse.cfm?elib=13948> (visité le 24/06/2015).
- [BDH10] Florent BERTHAUT, Myriam DESAINTE-CATHERINE et Martin HACHET. Drile : an immersive environment for hierarchical live-looping. In *New interface for musical expression*, 2010, page–192. URL : <https://hal.archives-ouvertes.fr/hal-00530071/> (visité le 04/11/2015).
- [BF93] Steve BENFORD et Lennart FAHLÉN. A spatial model of interaction in large virtual environments. In *Proceedings of the third european conference on computer-supported cooperative work 13–17 september 1993, milan, italy ECSCW'93*. Springer, 1993, pages 109–124.
- [Bre12] Jean BRESSON. Spatial structures programming for music. 2012) co-located with AAMAS (w21) :7, 2012. URL : <http://articles.ircam.fr/textes/Giavitto12c/index.pdf#page=17> (visité le 21/07/2015).
- [BV15] Jacob BEAL et Mirko VIROLI. Space-time programming. *Phil. trans. r. soc. a*, 373(2046) :20140220, 2015.
- [BW14] Mehul BHATT et Jan Oliver WALLGRÜN. Geospatial narratives and their spatio-temporal dynamics : commonsense reasoning for high-level analyses in geographic information systems. *ISPRS international journal of geo-information*, 3(1) :166–205, 2014. URL : <http://www.mdpi.com/2220-9964/3/1/166/htm> (visité le 02/11/2015).

- [CCLWOY15] Juan CHEN, Anthony G. COHN, Dayou LIU, Shengsheng WANG, Jihong OUYANG et Qiangyuan YU. A survey of qualitative spatial representations. *The knowledge engineering review*, 30(1) :106–136, 2015. URL : [http://journals.cambridge.org/abstract\\_S0269888913000350](http://journals.cambridge.org/abstract_S0269888913000350) (visité le 21/07/2015).
- [CD97] Eliseo CLEMENTINI et Paolino DI FELICE. A global framework for qualitative shape description. *GeoInformatica*, 1(1) :11–27, 1997. URL : <http://link.springer.com/article/10.1023/A:1009790715467> (visité le 02/11/2015).
- [CGPSB13] Julien COLAFRANCESCO, Pierre GUILLOT, Elliott PARIS, Anne SÈDES et Alain BONARDI. La bibliothèque hoa, bilan et perspectives. *Proc. of journées d'informatique musicale* :189–197, 2013. URL : [http://www.mshparisnord.fr/hoalibrary/wp-content/uploads/2013/05/La\\_bibliothèque\\_hoa\\_bilan\\_perspectives\\_jim2013.pdf](http://www.mshparisnord.fr/hoalibrary/wp-content/uploads/2013/05/La_bibliothèque_hoa_bilan_perspectives_jim2013.pdf) (visité le 22/07/2015).
- [Cha04] René CHALON. Réalité mixte et travail collaboratif : IRVO, un modèle de l’interaction homme-machine. Thèse de doctorat. Ecole Centrale de Lyon, 2004.
- [Che14] Zuofu CHENG. Design of a real-time GPU accelerated acoustic simulation engine for interactive applications. 00000. Thèse de doctorat. University of Illinois at Urbana-Champaign, 2014.
- [CMZHB99] Jonathan M COHEN, Lee MARKOSIAN, Robert C ZELEZNIK, John F HUGHES et Ronen BARZEL. An interface for sketching 3d curves. In *Proceedings of the 1999 symposium on interactive 3d graphics*. ACM, 1999, pages 17–21.
- [CTGH13] Dan CASAS, Margara TEJERA, J. GUILLEMAUT et Adrian HILTON. 4d parametric motion graphs for interactive animation. *Visualization and computer graphics, IEEE transactions on*, 19(5) :762–773, 2013. URL : [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=6365634](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6365634) (visité le 22/07/2015).
- [DEL04] M Olivier DELERUE. Spatialisation du son et programmation par contraintes : le système MusicSpace. Thèse de doctorat. UNIVERSITÉ PARIS 6, 2004.
- [DM14] Christopher H. DORR et Reinhard MORATZ. Qualitative shape representation based on the qualitative relative direction and distance calculus eOPRAM. *arXiv preprint arXiv:1412.6649*, 2014. URL : <http://arxiv.org/abs/1412.6649> (visité le 21/07/2015).
- [EGS08] David EPPSTEIN, Michael T GOODRICH et Jonathan Z SUN. Skip quadtrees : dynamic data structures for multidimensional point sets. *International journal of computational geometry & applications*, 18(1) :131–160, 2008.
- [EPSZC14] Emile ELLBERGER, Germán Toro PEREZ, Johannes SCHUETT, Giorgio ZOIA et Linda CAVALIERO. Spatialization symbolic music notation at ICST, 2014.

- [FBCG15] Dominique FOBER, Jean BRESSON, Pierre COUPRIE et Yann GESLIN. Les nouveaux espaces de la notation musicale. In *Journées d'informatique musicale*, 2015. URL : <https://hal.archives-ouvertes.fr/hal-01160759/> (visité le 02/11/2015).
- [FCEPSW98] Thomas FUNKHOUSER, Ingrid CARLBOM, Gary ELKO, Gopal PINGALI, Mohan SONDHI et Jim WEST. A beam tracing approach to acoustic modeling for interactive virtual environments. In *Proceedings of the 25th annual conference on computer graphics and interactive techniques*. ACM, 1998, pages 21–32. URL : <http://dl.acm.org/citation.cfm?id=280818> (visité le 12/04/2015).
- [FGB15] Xavier FAVORY, Jérémie GARCIA et Jean BRESSON. Trajectoires : une application mobile pour le contrôle et l'écriture de la spatialisation sonore, 2015. 00000. URL : <http://jeremiegarcia.fr/doc/traj-ihm15.pdf> (visité le 03/08/2015).
- [FTJ03] Thomas FUNKHOUSER, Nicolas TSINGOS et Jean-Marc JOT. Survey of methods for modeling sound propagation in interactive virtual environment systems, 2003.
- [GA12] Codruta GIRLEA et Eyal AMIR. Probabilistic region connection calculus. *Spatio-temporal dynamics* :62, 2012. URL : [http://www2.lirmm.fr/ecai2012/images/stories/ecai\\_doc/pdf/workshop/W1\\_STeDy-12-ECAI-Proceedings.pdf#page=78](http://www2.lirmm.fr/ecai2012/images/stories/ecai_doc/pdf/workshop/W1_STeDy-12-ECAI-Proceedings.pdf#page=78) (visité le 23/07/2015).
- [GBB10] Sean GUSTAFSON, Daniel BIERWIRTH et Patrick BAUDISCH. Imaginary interfaces : spatial interaction with empty hands and without visual feedback. In *Proceedings of the 23rd annual ACM symposium on user interface software and technology*. ACM, 2010, pages 3–12.
- [GBS03] Tovi GROSSMAN, Ravin BALAKRISHNAN et Karan SINGH. An interface for creating and manipulating curves using a high degree-of-freedom curve input device. In *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 2003, pages 185–192. URL : <http://dl.acm.org/citation.cfm?id=642645> (visité le 22/07/2015).
- [Gre03] Pierre GRENON. *The formal ontology of spatio-temporal reality and its formalization*. Amsterdam : AAAI Press, 2003. 27–34. URL : <http://www.aaai.org/Papers/Symposia/Spring/2003/SS-03-03/SS03-03-006.pdf> (visité le 03/11/2015).
- [Haz12] Shyamanta M. HAZARIKA, éditeur. Qualitative spatio-temporal representation and reasoning : trends and future directions. IGI Global, 2012. ISBN : 978-1-61692-868-1 978-1-61692-870-4. URL : <http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-61692-868-1> (visité le 12/04/2015).
- [HCNP03] Paul HUDAK, Antony COURTNEY, Henrik NILSSON et John PETERSON. Arrows, robots, and functional reactive programming. In, *Advanced functional programming*, pages 159–187. Springer, 2003. URL : [http://link.springer.com/content/pdf/10.1007/978-3-540-44833-4\\_6.pdf](http://link.springer.com/content/pdf/10.1007/978-3-540-44833-4_6.pdf) (visité le 22/07/2015).

- [HGHLB12] Horst HÖRTNER, Matthew GARDINER, Roland HARING, Christopher LINDINGER et Florian BERGER. *Spixels, pixels in space*, juin 2012. url : [http://www.researchgate.net/profile/Horst\\_Hoertner/publication/260823920\\_Spexels\\_Pixels\\_in\\_Space\\_A\\_novel\\_mode\\_of\\_spatial\\_display/links/0f31753270c857a9d8000000.pdf](http://www.researchgate.net/profile/Horst_Hoertner/publication/260823920_Spexels_Pixels_in_Space_A_novel_mode_of_spatial_display/links/0f31753270c857a9d8000000.pdf) (visité le 03/11/2015).
- [Hie05] Garin HIEBERT. Openal 1.1 specification and reference, 2005. 00010.
- [JBLFC14] Thibaut JACOB, Gilles BAILLY, Eric LECOLINET, Raphael FOULON et Etienne CORTEEL. A design space for three-dimensional curve edition. In. ACM Press, 2014, pages 105–112. isbn : 9781450329354. doi : 10.1145/2670444.2670454. url : <http://dl.acm.org/citation.cfm?doid=2670444.2670454> (visité le 22/07/2015).
- [JCR12] Guillaume JACQUEMIN, Thierry CODUYS et Matthieu RANC. Iannix 0.8. *Actes des journées d'informatique musicale (JIM 2012). mons (belgium) :107–15*, 2012. url : [http://jim.afim-asso.org/jim12/pdf/jim2012\\_18\\_p\\_jacquemin.pdf](http://jim.afim-asso.org/jim12/pdf/jim2012_18_p_jacquemin.pdf) (visité le 04/11/2015).
- [JH15] Jacek JANKOWSKI et Martin HACHET. Advances in interaction with 3d environments. In *Computer graphics forum*. Tome 34. 00001. Wiley Online Library, 2015, pages 152–190. url : <http://onlinelibrary.wiley.com/doi/10.1111/cgf.12466/full> (visité le 24/06/2015).
- [JRSJBS13] Jacek JANKOWSKI, Sandy RESSLER, Kristian SONS, Yvonne JUNG, Johannes BEHR et Philipp SLUSALLEK. Declarative integration of interactive 3d graphics into the world-wide web : principles, current approaches, and research agenda. In *Proceedings of the 18th international conference on 3d web technology*. In Web3D '13. ACM, New York, NY, USA, 2013, pages 39–45. isbn : 978-1-4503-2133-4. doi : 10.1145/2466533.2466547. url : <http://doi.acm.org/10.1145/2466533.2466547>.
- [JWJBF10] Yvonne JUNG, Sebastian WAGNER, Christoph JUNG, Johannes BEHR et Dieter FELLNER. Storyboarding and pre-visualization with x3d. In *Proceedings of the 15th international conference on web 3d technology*. ACM, 2010, pages 73–82. url : <http://dl.acm.org/citation.cfm?id=1836060> (visité le 04/11/2015).
- [KCGF14] Rubaiat Habib KAZI, Fanny CHEVALIER, Tovi GROSSMAN et George FITZMAURICE. Kitty : sketching dynamic and interactive illustrations. In *Proceedings of the 27th annual ACM symposium on user interface software and technology*. ACM, 2014, pages 395–405. url : <http://dl.acm.org/citation.cfm?id=2647375> (visité le 22/07/2015).
- [KCY93] Arie KAUFMAN, Daniel COHEN et Roni YAGEL. Volume graphics. *IEEE computer*, 26(7) :51–64, 1993.

- [Kim14] Chungyun KIM. Object-based spatial audio : concept, advantages, and challenges. In Ahmet KONDOZ et Tasos DAGIUKLAS, éditeurs, *3d future internet media*, pages 79–84. Springer New York, 2014. ISBN : 978-1-4614-8372-4. URL : [http://dx.doi.org/10.1007/978-1-4614-8373-1\\_5](http://dx.doi.org/10.1007/978-1-4614-8373-1_5).
- [KJGA06] M KALTENBRANNER, Sergi JORDA, Gunter GEIGER et Marcos ALONSO. The reactable : a collaborative musical instrument. In *Enabling technologies : infrastructure for collaborative enterprises*, 2006. WETICE'06. 15th IEEE international workshops on. IEEE, 2006, pages 406–411.
- [KL06] Mika KUUSKANKARE et Mikael LAURSON. Expressive notation package. *Computer music journal*, 30(4) :67–79, 2006.
- [KNL11] Jenny KIDD, Irida NTALLA et William LYONS. Multi-touch interfaces in museum spaces : reporting preliminary findings on the nature of interaction. *Rethinking technology in museums : emerging experiences. university of limerick*, 2011. URL : <http://www.idc.ul.ie/techmuseums11/paper/paper1.pdf> (visité le 02/11/2015).
- [KPGTB08] Gary S. KENDALL, Nils PETERS, Matthias GEIER, Deutsche TELEKOM et G. BERLIN. Towards an interchange format for spatial audio scenes. In *Proceedings of the international computer music conference*, 2008, pages 295–296. URL : [http://www.int.uni-rostock.de/fileadmin/user\\_upload/publications/st/mgeier/kendall2008interchange\\_format.pdf](http://www.int.uni-rostock.de/fileadmin/user_upload/publications/st/mgeier/kendall2008interchange_format.pdf) (visité le 22/07/2015).
- [LC09] Congcong LI et Tsuhan CHEN. Aesthetic visual quality assessment of paintings. *Selected topics in signal processing, IEEE journal of*, 3(2) :236–252, 2009.
- [Le 13] Cécile LE PRADO. Ecriture sonore : entre déterminisme, émergence et interactivité. Thèse de doctorat. Paris, CNAM, 2013.
- [LH13] Dongwook LEE et Minsoo HAHN. StageViz : visualization and storyboarding tool for performing art. In *Information science and applications (ICISA), 2013 international conference on*. IEEE, 2013, pages 1–3. URL : [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=6579494](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6579494) (visité le 12/04/2015).
- [LKH13] Dongwook LEE, Jinsul KIM et Minsoo HAHN. Visualization and authoring system for robot based performing art, 2013. URL : [http://onlinepresent.org/proceedings/vol42\\_2013/10.pdf](http://onlinepresent.org/proceedings/vol42_2013/10.pdf) (visité le 04/11/2015).
- [LKKH14] Dongwook LEE, Jinsul KIM, Ilmin KIM et Minsoo HAHN. Virtual assembly system for robot actor in performing art. *International journal of software engineering & its applications*, 8(6), 2014. URL : [http://www.sersc.org/journals/IJSEIA/vol18\\_no6\\_2014/26.pdf](http://www.sersc.org/journals/IJSEIA/vol18_no6_2014/26.pdf) (visité le 12/04/2015).

- [LL10] Jacques LEMORDANT et Yohan LASORSA. Augmented reality audio editing. In *Audio engineering society convention 128*. Audio Engineering Society, 2010. [url : http://www.aes.org/e-lib/online/browse.cfm?elib=15439](http://www.aes.org/e-lib/online/browse.cfm?elib=15439) (visité le 12/04/2015).
- [Lod14] Simona LODI. Spatial narratives in art. In, *Augmented reality art*, pages 277–294. Springer, 2014.
- [Mef13] Wafa MEFTEH. Approche ontologique pour la modélisation et le raisonnement sur les trajectoires : prise en compte des aspects thématiques, temporels et spatiaux. Thèse de doctorat. La Rochelle, 2013.
- [Mer06a] Bertrand MERLIER. *Vocabulaire de l'espace en musiques électroacoustiques*, 2006.
- [Mer06b] Bertrand MERLIER. VOCABULAIRE DE L'ESPACE ET DE LA SPATIALISATION DES MUSIQUES ÉLECTROACOUSTIQUES : PRÉSENTATION, PROBLÉMATIQUE ET TAXINOMIE DE L'ESPACE. In *EMS : electroacoustic music studies network-beijing 2006*, 2006, page 247. [url : https://halshs.archives-ouvertes.fr/halshs-00521322/](https://halshs.archives-ouvertes.fr/halshs-00521322/) (visité le 03/11/2015).
- [MG11] Kevin MATLAGE et Andy GILL. Every animation should have a beginning, a middle, and an end. In, *Trends in functional programming*, pages 150–165. Springer, 2011. [url : http://link.springer.com/chapter/10.1007/978-3-642-22941-1\\_10](http://link.springer.com/chapter/10.1007/978-3-642-22941-1_10) (visité le 02/11/2015).
- [MLV05] F. MELCHIOR, T. LAUBACH et D. de VRIES. Authoring and user interaction for the production of wave field synthesis content in an augmented reality system. In. IEEE, 2005, pages 48–51. ISBN : 978-0-7695-2459-7. DOI : 10.1109/ISMAR.2005.20. [url : http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1544662](http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1544662) (visité le 02/11/2015).
- [MPCZHC10] Despina MICHAEL, Nectarios PELEKANOS, Isabelle CHRYSANTHOU, Panagiotis ZAHARIAS, Loukia L. HADJIGAVRIEL et Yiorgos CHRYSANTHOU. Comparative study of interactive systems in a museum. In, *Digital heritage*, pages 250–261. Springer, 2010. [url : http://link.springer.com/chapter/10.1007/978-3-642-16873-4\\_19](http://link.springer.com/chapter/10.1007/978-3-642-16873-4_19) (visité le 12/04/2015).
- [NMSH03] Markus NOISTERNIG, Thomas MUSIL, Alois SONTACCHI et Robert HÖLDRICH. 3d binaural sound reproduction using a virtual ambisonic approach. In *Virtual environments, human-computer interfaces and measurement systems, 2003. VECIMS'03. 2003 IEEE international symposium on*. IEEE, 2003, pages 174–178.
- [NS14] Luiz NAVEDA et Ivani SANTANA. “topos” toolkit for pure data : exploring the spatial features of dance gestures for interactive musical applications, 2014. [url : http://www.researchgate.net/profile/Luiz\\_Naveda2/publication/266141223\\_Topol\\_toolkit\\_for\\_Pure\\_Data\\_exploring\\_the\\_spatial\\_features\\_of\\_dance\\_gestures\\_](http://www.researchgate.net/profile/Luiz_Naveda2/publication/266141223_Topol_toolkit_for_Pure_Data_exploring_the_spatial_features_of_dance_gestures_)

- for\_interactive\_musical\_applications/links/5426ca440cf2e4ce940a1916.pdf (visité le 21/07/2015).
- [Per15] Andres PEREZ-LOPEZ. 3dj - supercollider framework for live sound spatialization. Linux Audio Conference, 2015.
- [PLS13] Nils PETERS, Trond Lossius et Jan C. SCHACHER. The spatial sound description interchange format : principles, specification, and examples. *Computer music journal*, 37(1) :11–22, 2013. URL : [http://www.mitpressjournals.org/doi/abs/10.1162/COMJ\\_a\\_00167](http://www.mitpressjournals.org/doi/abs/10.1162/COMJ_a_00167) (visité le 12/04/2015).
- [PLSBBP09] Nils PETERS, Trond Lossius, Jan SCHACHER, Pascal BALTAZAR, Charles BASCOU et Timothy PLACE. A stratified approach for sound spatialization. In *Proceedings of 6th sound and music computing conference*, 2009, pages 219–224. URL : <http://spatdif.org/papers/Spatialization-SMC09.pdf> (visité le 04/11/2015).
- [PLV08] Bruce PORTER, Vladimir LIFSHITZ et Frank VAN HARMELEN, éditeurs. Handbook of knowledge representation. 1st ed édition. In Foundations of artificial intelligence. Elsevier, Amsterdam ; Boston, 1st ed édition, 2008. 1005 pages. ISBN : 9780444522115 0444522115.
- [Ras13] Joakim RASMUSON. Flying colors : toward more natural and intuitive 3d modeling, 2013. URL : <http://www.diva-portal.org/smash/record.jsf?pid=diva2:837642> (visité le 04/11/2015).
- [Rod14] Jose Antonio Belloch RODRIGUEZ. Performance improvement of multichannel audio by graphics processing units, 2014. URL : <http://hgpu.org/?p=12916> (visité le 12/04/2015).
- [RS14] Nikunj RAGHUVANSHI et John SNYDER. Parametric wave field coding for precomputed sound propagation. *ACM transactions on graphics*, 33(4) :1–11, 27 juillet 2014. ISSN : 07300301. DOI : 10.1145/2601097.2601184. URL : <http://dl.acm.org/citation.cfm?doid=2601097.2601184> (visité le 12/04/2015).
- [Sal04] Liliana SALAZAR. Modélisation et analyse spatiale et temporelle des jeux vidéo basées sur les réseaux de pétri. Thèse de doctorat. CNAM, 2004.
- [SCBS13] Stephanie SANTOSA, Fanny CHEVALIER, Ravin BALAKRISHNAN et Karan SINGH. Direct space-time trajectory control for visual media editing. In *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 2013, pages 1149–1158. URL : <http://dl.acm.org/citation.cfm?id=2466148> (visité le 22/07/2015).
- [Sch96] Christoph SCHLIEDER. Qualitative shape representation. *Geographic objects with indeterminate boundaries*, 2 :123–140, 1996.
- [SCV13] Yuya SASAMOTO, Michael COHEN et Julian VILLEGRAS. Controlling spatial sound with table-top interface. In *Awareness science and technology and ubi-media computing (iCAST-UMEDIA), 2013 international joint conference on*. IEEE, 2013, pages 713–718.

- [SD13] Jeremy SCOTT et Randall DAVIS. Physink : sketching physical behavior. In. ACM Press, 2013, pages 9–10. ISBN : 9781450324069. doi : 10.1145/2508468.2514930. URL : <http://dl.acm.org/citation.cfm?doid=2508468.2514930> (visité le 21/07/2015).
- [SL14] Yang Ting SHEN et Pei Wen Lu. BlowBrush : a design of tangible painting system using blowing action. In, *Distributed, ambient, and pervasive interactions*, pages 184–195. Springer, 2014.
- [Spr11] Michael SPRANGER. Recruitment, selection and alignment of spatial language strategies. In *Advances in artificial life, ECAL 2011 : proceedings of the eleventh european conference on the synthesis and simulation of living systems*, 2011. URL : <https://mitp-web2.mit.edu/sites/default/files/titles/alife/0262297140chap116.pdf> (visité le 03/11/2015).
- [SS12] Michael SPRANGER et Luc STEELS. Emergent functional grammar for space. *Experiments in cultural language evolution*. amsterdam : john benjamins. <http://dx.doi.org/10.1075/ais.3>, 2012. URL : <https://www.cs1.sony.fr/downloads/papers/2012/spranger-12c.pdf> (visité le 03/11/2015).
- [SSJGB04] James SHERIDAN, Gaurav SOOD, Thomas JACOB, Henry GARDNER et Stephen BARRASS. Soundstudio 4d : a VR interface for gestural composition of spatial soundscapes. In *ICAD*, 2004.
- [SYKK13] Ju-Hwan SEO, Jeong-Yean YANG, JaeWoo KIM et Dong-Soo KWON. Autonomous humanoid robot dance generation system based on real-time music input. In *RO-MAN, 2013 IEEE*. IEEE, 2013, pages 204–209.
- [TCMLSM12] Micah TAYLOR, Anish CHANDAK, Qi Mo, Christian LAUTERBACH, Carl SCHISSLER et Dinesh MANOCHA. Guided multiview ray tracing for fast auralization. *Visualization and computer graphics, IEEE transactions on*, 18(11) :1797–1810, 2012.
- [TG98] Nicolas TSINGOS et Jean-Dominique GASCUEL. Fast rendering of sound occlusion and diffraction effects for virtual acoustic environments. In *Audio engineering society convention 104*. Audio Engineering Society, 1998. URL : <http://www.aes.org/e-lib/browse.cfm?elib=8481> (visité le 21/07/2015).
- [TL14] Titus J. J. TANG et Wai Ho LI. An assistive EyeWear prototype that interactively converts 3d object locations into spatial audio. In. ACM Press, 2014, pages 119–126. ISBN : 9781450329699. doi : 10.1145/2634317.2634318. URL : <http://dl.acm.org/citation.cfm?doid=2634317.2634318> (visité le 12/04/2015).
- [VW06] Doug VAN NORT et Marcelo WANDERLEY. The LoM mapping toolbox for max/MSP/jitter. In *Proc. of the 2006 international computer music conference (ICMC)*, 2006, pages 397–400.

- [WBDO14] David WAGNER, Ludger BRÜMMER, Götz DIPPER et Jochen Arne OTTO. Introducing the zirkonium MK2 system for spatial composition, 2014. URL : <http://www.smc-conference.net/smc-icmc-2014/images/proceedings/OS16-B05-IntroducingtheZirkonium.pdf> (visité le 22/07/2015).
- [Woz06] Michael WOZNIEWSKI. A framework for interactive three-dimensional sound and spatial audio processing in a virtual environment. Thèse de doctorat. Citeseer, 2006.
- [WQS12] Mike WOZNIEWSKI, Alexandre QUESSY et Zack SETTEL. SPATOSC : PROVIDING ABSTRACTION FOR THE AUTHORIZING OF INTERACTIVE SPATIAL SUDIO EXPERIENCES. Ann Arbor, MI : MPublishing, University of Michigan Library, 2012.
- [WS11] Graham WAKEFIELD et Wesley SMITH. *Cosm : a toolkit for composing immersive audio-visual worlds of agency and autonomy*. Ann Arbor, MI : MPublishing, University of Michigan Library, 2011. URL : [http://www.mat.ucsb.edu/~wakefield/pubs/11-WakefieldSmith\\_ICMC2011\\_Cosm.pdf](http://www.mat.ucsb.edu/~wakefield/pubs/11-WakefieldSmith_ICMC2011_Cosm.pdf) (visité le 24/06/2015). 00007.