

OUTILS D'ÉCRITURE SPATIALE POUR LES PARTITIONS INTERACTIVES

Jean-Michel Celerier

Univ. Bordeaux, LaBRI, UMR 5800, F-33400 Talence, France.

Blue Yeti, F-17110 France.

jcelerie@labri.fr

Myriam Desainte-Catherine

Univ. Bordeaux, LaBRI, UMR 5800, F-33400 Talence, France.

CNRS, LaBRI, UMR 5800, F-33400 Talence, France.

INRIA, F-33400 Talence, France.

myriam@labri.fr

Jean-Michel Couturier

Blue Yeti, F-17110 France.

jmc@blueyeti.fr

RÉSUMÉ

L'écriture de partitions interactives vise à inclure du contenu spatial. En effet, dans de nombreuses pratiques de création, musicales ou scénographiques, un travail sur des paramètres plusieurs dimensions apparaît nécessaire. Cela peut être dû à la spatialisation du son ou à la nécessité de répondre des contraintes physiques sur une scène. Il est possible de définir des sous-espaces utiles pour la définition de ces paramètres en tant que zones d'espace : les courbes de trajectoires paramétriques en sont un exemple. Nous présentons les modèles possibles pour l'écriture de contenu spatial. Un modèle mixte est choisi pour l'implémentation dans le séquenceur i-score. Il offre des moyens d'écriture très généraux ainsi que la possibilité de spécialiser et d'optimiser certains cas lorsque cela s'avère nécessaire. Cette implémentation comporte des outils d'édition, de visualisation, ainsi que deux possibilités sémantiques d'exécution offrant une forte interopérabilité avec les scénarios interactifs.

1. INTRODUCTION

Les pratiques de création multimedia incluent par définition des composantes spatiales. Celles-ci peuvent se manifester dès qu'un des types de données manipulées possède plus d'une dimension. La simple présence de multiples paramètres dans une création multimedia permet d'avoir une notion d'espace dans lequel on met en correspondance un paramètre avec un autre. Il est donc possible de définir de manière géométrique de nombreux éléments courants avec lesquels les compositeurs travaillent.

Cependant, la plupart des outils permettant un travail d'écriture spatiale sont soit spécifiques à la création de trajectoires pour des sons, dans un cadre musical, soit orientés vers la création de jeu vidéo ou de formes spécifiques de médias. De par leur spécialisation pour des projets donnés, ils ne permettent pas toujours une utilisation facile dans un contexte autre que celui qui a été prévu au départ.

Nous avons ici la démarche inverse : nous cherchons des méthodes générales, permettant de manipuler tout type de données spatiales pour ensuite pouvoir les appliquer aux

cas particuliers de la création de scénographies interactives avec des éléments musicaux. Cela a pour avantage une plus grande flexibilité, et une moindre nécessité d'utilisation de différents outils. Cependant, c'est au prix de performances non optimales, que nous essayons néanmoins d'optimiser par la suite.

Sera exposé un état de l'art des outils de création spatiale, puis nous présenterons deux modèles, l'un géométrique et l'autre paramétrique, qui permettent de réaliser différents types de compositions, notamment liées aux relations que l'on peut avoir entre plusieurs zones définies géométriquement. Ces modèles, intégrés au séquenceur i-score, sont appliqués deux études de cas : une chorégraphie de robots Metabots, et une application de réalité augmentée audio, Sono-pluie.

Des détails et problématiques propres à l'implémentation sont présentés, ainsi que les modes de manipulation et de rendu. Notamment, l'intégration d'un processus spatial à la sémantique d'exécution actuelle de i-score est discutée.

2. EXISTANT

Nous présentons ici plusieurs applications opérant principalement dans un domaine spatial, mais créatif. Les thèmes étudiés sont d'abord l'utilisation de contenus spatiaux en musique, en scénographie, et en jeux vidéo, ainsi que les différentes méthodes d'écriture et de représentation qui existent.

2.1. Contexte

Les développements réalisés sont intégrés au séquenceur i-score et en concordance avec les études menées sur l'interopérabilité durant les projets ANR Virage et OSSIA. Le projet OSSIA a abouti à la création de modèles graphiques et opérationnels pour les partitions interactives, ainsi que d'une interface de programmation ayant pour but de permettre l'écriture de scénarios interactifs, et spécifier la communication avec les objets et logiciels avec lesquels i-score interagit.

i-score est basé selon une architecture modulaire basée sur des plug-ins écrits en C++. Les processus temporels, qui

dcrivent les oprations et messages qui vont tre envoys chaque tick d'horloge, peuvent tre implments en tant que plug-ins. Il est donc ais d'en rajouter pour avoir de nouveaux comportements dans le logiciel, l'dition comme l'excution d'un scario. C'est l'approche qui est suivie pour les outils prsents dans cet article.

2.2. Partitions musicales spatiales et graphiques

Une prsentation exhaustive de l'tat actuel de l'criture spatiale en musique est donne dans [16]. Notamment, la question de la notation dans le cadre de partitions impliquant des lments spatiaux y est aborde.

Ces partitions, graphiques, peuvent tre spatiales uniquement dans leur reprsentation, mais peuvent aussi indiquer des manires d'interprter dans l'espace, notamment l'aide de symboles spcialiss [15]. Une taxinomie des possibilit de cration dans l'espace en musiques lectro-acoustiques est prsente par Bertrand Merlier, notamment dans l'ouvrage *Vocabulaire de l'espace en musique lectro-acoustique* [25].

Des outils logiciels existent pour ces partitions – ils sont souvent spcialiss. Par exemple, la bibliothque ENP [23] permet de concevoir des partitions graphiques l'aide d'un diteur lui aussi graphique et d'un langage bas sur LISP.

Une des problmatiques actuelles pour la reprsentation de l'criture musicale est celle du geste, et de son lien avec la partition : comment annoter le geste du musicien avec prcision ? Et, inversement, comment partir d'un geste crer un son correspondant ? Ces questions sont abordes dans la description de Soundstudio 4D [31], dans le cadre d'un systme de conception de trajectoires pour spatialisation l'aide d'interactions en trois dimensions.

Une autre question est l'association entre l'aspect graphique et le rsultat. Ainsi, des outils tels que Holo-Edit et Holo-Spat [9] permettent de travailler avec des trajectoires, mais ciblent spcifiquement la manipulation et spatialisation du son. C'est d la ncessit de composer en ayant conscience chaque instant des fortes contraintes techniques du moyen de restitution de l'œuvre ; par exemple, la configuration de hauts-parleurs pourra tre fixe par l'auditorium et l'artiste doit alors travailler avec.

Le logiciel IanniX [20] (fig. 1) dispose aussi de nombreuses possibilit d'criture spatiale : les partitions sont des ensembles d'lments graphiques dfinis paramtriquement ou bien l'aide d'un langage de programmation ddi, que des curseurs vont parcourir. L'information de position de chaque curseur est envoye en OSC, ce qui permet l'intgration d'autres logiciels, ainsi que des compositions par rtroaction : les paramtres mis par OSC peuvent modifier la structure du scario durant l'excution.

2.3. Modlisation de l'espace

Une mthode d'criture de la spatialisation par contraintes est propose par Olivier Delerue dans MusicSpace [13]. Cela permet une approche dclarative de l'criture de partition, en spcifiant des contraintes telles que « deux objets ne doivent jamais tre plus de deux mtres l'un de l'autre »

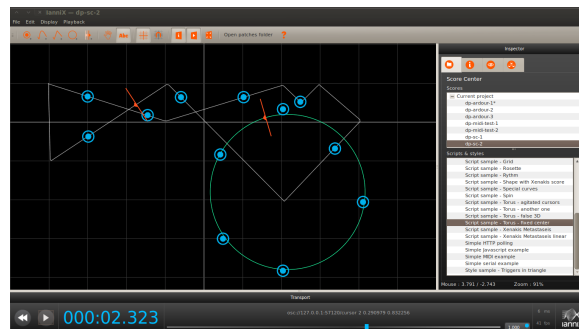


Figure 1. Le squenceur graphique IanniX, inspir de l'œuvre de Iannis Xenakis.

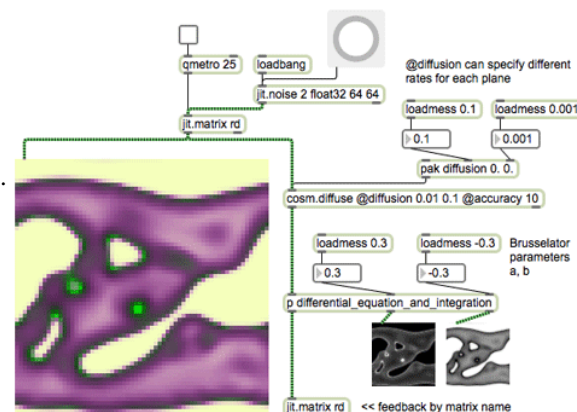


Figure 2. Un patch COSM dans Max/MSP.

ou bien « l'angle entre deux objets et l'auditeur doit tre suprieur 90 degrs ». Les objets peuvent tre notamment des sources sonores. Une dition graphique de ces contraintes est propose, et elles sont reprsentes en termes de cercles et de segments reliant les objets qu'elles contraignent.

Des donnes spatiales peuvent aussi tre utilises directement pour crer des mises en correspondance (*mappings*) sonores. C'est le cas de la bibliothque Topos [26], qui permet de capter le mouvement de danseurs et d'en extraire des informations pouvant tre utiles pour la conception de pices de musique interactives. Une fois que le mouvement du danseur est captur via un priphrique externe, il devient possible d'extraire des informations telles que le volume occup par le danseur, sa vitesse, ou bien diverses mesures relatives l'volution de deux ou quatres points dans le temps, comme l'instabilit ou les collisions entre diffrentes parties du corps. Ces donnes peuvent ensuite tre rutilises dans Pure Data pour de la gnration de musique.

Un modle plus complet d'espace en trois dimensions est fourni par COSM [34], en fig. 2. Implment dans Max/MSP, il offre une grande richesse d'criture mais n'a pas t mis jour pour les dernires versions de Max. En plus de lieux et de trajectoires, il est possible d'crire l'interaction dans une certaine mesure, ainsi que la communication entre diffrents agents. Une nouveaut est la possibilit de travailler avec des champs dfinis mathmatiquement. Ces champs peuvent varier dans le temps et tre sonifis par la suite.

Une approche de contrôle spatial est possible via le logiciel Blender, qui sert l'origine à réaliser des images et films de synthèse. Blender a déjà été utilisé d'autres occasions pour de la spatialisation de son [28] et de la simulation acoustique [8]. Blender peut être contrôlé via une API Python et il est notamment possible de déplacer des éléments et tester la présence de collisions ou d'autres propriétés géométriques. Néanmoins, cela se fait à la vitesse de son moteur d'exécution qui est fixé à 60 Hz. Les messages reçus entre deux trames sont accumulés.

Le monde des jeux vidéo dispose aussi d'outils adaptés à l'écriture spatiale : l'interface OpenAL a été développée l'origine pour offrir aux jeux une couche d'abstraction permettant de bénéficier de spatialisation simplement en donnant une position et une orientation des sources sonores ponctuelles. Cette position et orientation peuvent varier dans le temps. Par la suite les implémentations sont libres d'utiliser les méthodes qu'elles souhaitent pour réaliser la spatialisation : cela peut aller d'une simple panoramique gauche-droite à l'utilisation de HRTFs, comme le fait la bibliothèque OpenAL-soft.

De manière générale, les moteurs et éditeurs de jeux, tels que Unreal Engine ou Unity offrent des possibilités d'interaction très riches pour disposer et animer ces sources dans le temps.

2.4. Interfaces homme-machine pour le spatial

Enfin, il existe de nombreuses alternatives aux traditionnels claviers, souris, joysticks pour les modes et périphériques d'entrée et d'interaction permettant de décrire des données spatiales dans le monde physique. Par exemple, il existe plusieurs possibilités de composition musicale à l'aide de tables interactives comme la Reactable [22] et différentes approches drives qui peuvent être spécifiquement axées sur la spatialisation du son [29]. On notera aussi ShapeTape [18], un ruban déformable qui permet d'obtenir sa torsion en 32 points pour reconstruire la courbe en 3D dans un logiciel. Un vocabulaire de mouvements possibles (torsions, tirements, pousses, ...) est défini par les auteurs.

3. CAS D'UTILISATION

Afin d'avoir des possibilités de test de notre système, deux cas pratiques d'applications nécessitant de l'écriture de contenus spatiaux ont été étudiés.

Dans les deux cas, on veut créer une famille de scénarios interactifs qui manipulent des types d'objets fixes : des sources sonores virtuelles dans le premier cas et des robots dans le second cas.

En concordance avec les objectifs d'interopérabilité de l'environnement i-score, le protocole Minuit [12] est utilisé pour les cas présents. Il s'agit d'un protocole de communication basé sur la norme OSC permettant la découverte via réseau des paramètres contrôlables d'un logiciel multimedia. Les paramètres sont organisés en arbre, qui correspondent au modèle d'objet de l'application distante. Un exemple d'un tel arbre est montré en fig. 3.

```
/audio
  /volume [float : 0 - 1]
  /pan     [int : -100 - 100]
/video
  /play
  /pause
  /color   [float : 0 - 360]
```

Figure 3. Un exemple d'arbre Minuit pour une application simple. On place entre crochets le type et les bornes acceptées du paramètre. Les messages /video/play et /video/pause sont semblables des appels de fonction.

L'utilisation de Minuit nous permet d'accéder automatiquement aux paramètres des applications d'exemple depuis i-score, puis de les scénariser.

3.1. Exemple : Sonopluie

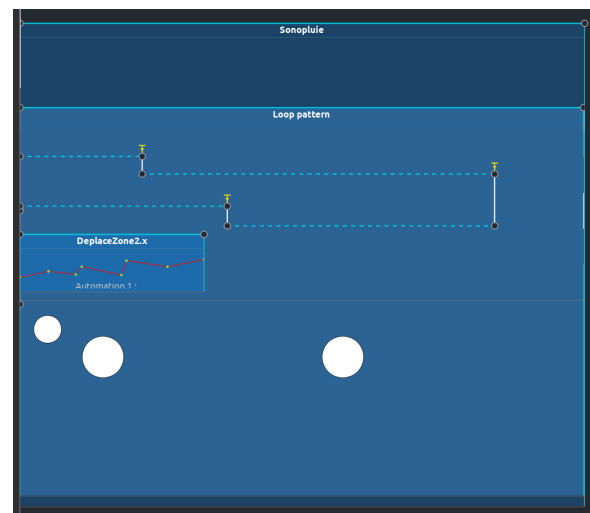


Figure 4. Un scénario de type Sonopluie

Cette application interactive utilise de la géolocalisation par téléphone. Plusieurs sources sonores sont apposées dans un espace, dans lequel les participants se déplacent. Cet espace virtuel est mis en correspondance avec le monde réel lors du parcours. La mise en correspondance est réalisée en positionnant les zones sur une carte issue d'un service web. Le système mesure la distance de chaque participant aux sources, et va jouer les sources sonores plus ou moins fort en fonction de cette distance.

Il manque un outil simple d'écriture pour disposer les zones dans l'espace, leur associer des sons et des comportements. Un des objectifs du projet est d'offrir de l'interaction et de l'évolativité dans les scénarios : par exemple, après s'être approché d'un point donné, on voudrait pouvoir désactiver des points précédents et activer des points suivants, ou avoir des points qui se déplacent.

Un scénario d'exemple, tel qu'implémenté dans i-score, consiste en :

- Une source de golocalisation expose dans un arbre Minuit.
- Un moteur audio grant la spatialisation via positionnement de sources. Nous utilisons ici la bibliothèque OpenAL [19], dans son implmentation OpenAL-soft. En effet, elle est supporte sur de nombreux appareils mobiles courants. Elle aussi est implmente en Minuit.
- Un projet i-score qui consiste en :
 - Une boucle principale contenant l'application complte.
 - Dans cette boucle, un processus Espace o sont instancies les zones sous forme de disque. Pour cet exemple il y a trois zones. Ce processus est dtaill dans la partie suivante.
 - Un nombre de pointeurs statiques est allou l'avance. Leur position correspond celle des sources de golocalisation.
 - On va raliser deux tests : distance au centre d'une zone, collision. La collision va servir activer le son, et la distance contrler un dosage d'effet, par exemple une rverbration.
 - Cette boucle contient aussi un scnario muni de plusieurs contraintes temporelles. Ces contraintes ont pour mission d'animer une des zones en la dplaant au cours du temps, et d'activer la troisieme zone uniquement quand les deux premieres ont t visites.

Ce scnario est visible en figure 4.

3.2. Exemple : robots

Ce projet consiste en la ralisation d'une chorgraphie de robots musicaux, puis de drones. On dispose d'une flotte de robots open-source Metabot ¹. Ces robots sont contrls en vitesse : on crit sur le port srie des commandes telles que $dx\ 5$ pour indiquer une vitesse de 5 cm s^{-1} . Un logiciel a t conu pour faire une mise en correspondance d'un arbre Minuit vers le port srie. En parallle, un logiciel de simulation, prsent en fig. 5 est dvelopp l'aide d'OpenFrameworks, qui expose le mme arbre Minuit. Ce logiciel de simulation permet d'afficher les robots et de dtecter les collisions, pour empcher qu'elles ne se produisent en pratique, avec des robots coteux. On veut notamment utiliser le rythme de marche des robots pour produire de la musique.

Chaque robot va possder une trajectoire dans le plan. Cette trajectoire pourra potentiellement changer en fonction des choix du metteur en scne, ou bien du comportement des autres robots : par exemple, si un robot tombe en panne en face d'un autre, un changement de trajectoire doit tre opr (ou bien le robot doit s'arrter). Pour ce faire, un autre logiciel est prsent sur le rseau : ce logiciel dtecte chaque instant la position de chaque robot et l'expose via Minuit. L'auteur peut donc utiliser cette information pour prvoir des cas d'urgence, comme mettre fin au spectacle gracieusement si trop de robots tombent en panne.

1. <http://metabot.fr/>

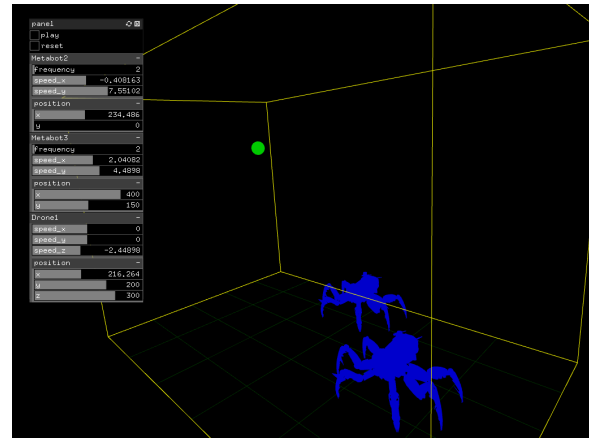


Figure 5. Logiciel de simulation de metabots communiquant avec i-score

Un des problmes principaux est le contrle en vitesse des robots : il faut partir d'une trajectoire dcrivant une position, calculer la drive dans les deux ou trois directions pour l'envoyer aux robots. Dans l'attente d'un modle de calcul plus complet et gnrique, cela peut tre fait dans le processus qui gre des trajectoires.

4. MODLE

4.1. Conception

Les objectifs pour l'intgration des notions spatiales dans i-score sont :

- Une intgration avec l'cosystme existant.
- Support de l'animation : Une description fine des volutions dans le temps (ou en fonction d'autres paramtres).
- La possibilit d'crire des scnes 2D, 3D basiques potentiellement en utilisant des guides (sources cartographiques, images).
- Des performances les plus proches possible du temps rel.

4.1.1. Modles pour l'criture spatiale

Plusieurs modles ont t envisags pour crire des scnarios spatiaux.

Une premiere possibilit repose sur les mthodes de description qualitative de l'espace, telles que RCC-8 [17]. Elles consistent en l'extension des relations de Allen [2], plus d'une dimension. Cependant de telles mthodes ne s'avrent pas efficaces si on dsire dcrire des dispositions prcises, avec des mtriques sur les objets que l'on veut manipuler. Par exemple, on peut l'aide de mthodes qualitatives crire "A est l'extrieur de B et lui est tangent, et A et B sont contenus dans C". Mais les pratiques appellent plus souvent une criture de la forme "A est un cercle de $1u$ centr en $(3, 3)$ dans une pice de $5u$, et un pointeur B est situ en $(2, 3)$ " ou u est une unit de distance ; on peut par la suite obtenir les relations RCC partir des relations mtriques tandis que l'inverse n'est pas possible.

Une seconde possibilité est l'approche orientée objet [32] : on dispose d'une classe abstraite représentant une forme. Les classes dérivées implémentent des méthodes de dessin et de calcul optimisées pour des formes concrètes : sphère, cube, etc. L'avantage est la possibilité d'avoir des exécutions très rapides dans certains cas. Par exemple, calculer la distance entre deux cercles est trivial. En revanche, cette méthode manque doublement de flexibilité :

- Il est difficile de rajouter de nouveaux types de calculs, car il faut modifier tous les types pour rajouter le calcul (par exemple un calcul d'aire pour une zone).
- Si un calcul implique deux (ou plus) zones abstraites, on assiste à une explosion combinatoire lorsque l'on désire rajouter une nouvelle zone car le calcul doit alors être défini entre la nouvelle zone et toutes les zones pré-existantes.

Une troisième possibilité, numérique, et permettant un maximum de flexibilité, a été étudiée : on demande à l'utilisateur de remplir la fonction caractéristique d'une zone à l'aide d'un langage de programmation. Par la suite, on peut tester pour chaque point de l'espace son appartenance à la zone, en ayant la possibilité de faire des approximations. Cela permet d'implémenter des objets spatiaux qui sont difficilement réalisables uniquement de manière mathématique, avec des objets dont la définition contient des boucles ou des conditionnelles, voire des facteurs aléatoires. En revanche, on perd des possibilités d'analyse par la suite : on n'a en effet pas de forme générale des zones ainsi créées et on ne peut au mieux qu'effectuer des tests, coûteux en temps de calcul, entre zones. Il est cependant toujours possible de prendre un échantillon des points de la zone pour appliquer une méthode de maillage telle que la triangulation de Delaunay [27] qui produit un objet sur lequel on peut réaliser plusieurs opérations plus simplement. Les bibliothèques CGAL [6] et VTK [30] proposent toutes deux des implémentations performantes de cette triangulation. Enfin, les performances ne sont pas adaptées à des évolutions rapides en temps réel : c'est le cas de la géométrie dynamique. Cela peut marcher en dimension 1 et pour des fonctions avec un nombre relativement important de points à calculer, mais est lourd en ressources en dimension 2 et quasiment impossible sur un CPU en temps réel en dimension 3.

Finalement, l'utilisation de modèles mathématiques hybrides est possible. En général, il s'agit des modèles paramétriques, ou géométriques. Les modèles géométriques sont notamment très utilisés dans le domaine de la conception assistée par ordinateur, tandis que les modèles paramétriques sous-tendent la plupart des outils de spatialisation musicale, notamment à l'aide des fonctions spline, ainsi que de nombreux outils pour l'animation et les images de synthèse.

4.1.2. Dimensions non spatiales

Une question qui se pose pour le choix du modèle est celle de l'interaction entre les zones et le temps. En effet, dans des logiciels comme IanniX, on dispose de courbes paramétrées par des paramètres externes ou par une horloge. Cette approche permet d'avoir dans une seule vue toute

l'information possible : on dessine la courbe pour un intervalle inclus dans l'ensemble de paramétrisation. Cependant, l'objectif des processus spatiaux est de posséder plus d'une dimension de paramétrisation. La question de l'affichage d'une variation au cours du temps se pose alors. On peut penser à afficher un dégradé qui serait complètement opaque pour une valeur de paramètre donnée, et serait de plus en plus transparent lorsque l'on s'en éloigne ; un exemple d'affichage d'animation d'objet 3D au cours du temps est par exemple donné dans [10]. Il est aussi possible d'afficher uniquement la trajectoire du centre (ou d'un point donné) de l'objet au cours du temps, mais cela nécessite de pouvoir la calculer. Ce n'est pas possible dans le cas d'une zone définie avec des contraintes très faibles. Par exemple, pour un modèle géométrique, défini par $x < 0$, un demi-plan, on ne peut le définir.

Une autre possibilité est d'avoir des boîtes sœurs pour définir les animations. Cela permet plus de clarté, et autorise d'associer une seule trajectoire à plusieurs objets plus facilement : La boîte contenant la trajectoire va écrire chaque tic d'horloge sur une adresse OSC. Les zones spatiales peuvent ensuite aller chercher la valeur de cette adresse ; l'auteur peut utiliser ces coordonnées dans l'écriture de l'interactivité de ces zones.

4.1.3. Espaces non cartésiens

Une considération importante est le support des objets définis autrement que dans le système de coordonnées cartésiennes. Le système le plus courant sera les coordonnées polaires, mais la question des espaces sur des types de paramètres autres a aussi été posée : par exemple, un travail sur des espaces colorimétriques requerrait des outils adaptés car la conversion entre plusieurs de ces espaces, comme RVB et $L^*a^*b^*$ est souvent non-linéaire. Pour ce faire, il est possible d'utiliser des outils tels que ceux fournis par la bibliothèque Jamoma [12], offrant notamment des types de données de haut-niveau dans la *DataspaceLib*.

4.2. Processus spatiaux dans i-score

Nous choisissons d'utiliser plusieurs des modèles exposés précédemment. Le cœur de l'approche repose sur un modèle mixte, géométrique et orienté objet pour la définition de scènes spatiales, ainsi que sur des processus représentant des objets paramétriques, en raison de leur praticité pour l'écriture de trajectoires, très courantes dans les domaines d'application cibles.

Il n'y a pas de focalisation directe sur le son : les processus opèrent toujours sur des données numériques quelconques.

4.2.1. Processus Espace

Dans le premier cas, on définit une classe abstraite de zone, qui contient une liste d'équations et d'inéquations. Ces équations et inéquations servent à contraindre la zone. Par exemple, on peut donner l'équation cartésienne d'un cercle, ou d'une parabole.

De 1, il est possible d'offrir un comportement gnrique pour l'affichage et les relations entre plusieurs zones, mais il est aussi possible de spcialiser la classe de zone des fins d'optimisations.

Ces zones sont contenues dans un processus Espace. Le processus Espace consiste en :

- Une liste de dimensions bornes.
- Une liste de zones.
- Une liste d'oprations entre zones.
- Une fenetre (*viewport*).

Lorsque la dfinition d'une zone est donnee, on spare en deux ses inconnues. Pour les exemples qui suivent, on se place dans un espace deux dimensions x, y .

Par exemple, dans le cas de l'quation d'un disque :

$$(u - x_0)^2 + (v - y_0)^2 \leq r^2 \quad (1)$$

On permet l'utilisateur d'attribuer chaque inconnue, u, v, x_0, y_0, r , soit une dimension, soit un paramtre. Les zones prdfinies par hritage offrent une dcomposition simple par dfaut. Ainsi, pour le disque, on utilise comme dimensions :

$$\begin{cases} x \rightarrow u \\ y \rightarrow v \end{cases} \quad (2)$$

L'utilisateur peut par la suite utiliser pour les paramtres une application de la forme :

$$\begin{cases} 10.0 \rightarrow x_0 \\ 10.0 \rightarrow y_0 \\ \text{/adresse/OSC} \rightarrow r_0 \end{cases} \quad (3)$$

Actuellement, les primitives prdfinies sont disque et pointeur. Le pointeur correspond :

$$\begin{cases} u = x_0 \\ v = y_0 \end{cases} \quad (4)$$

Sont prvues par la suite des implmentations spcifiques au moins pour les droites, segments, plans et demi-plans, polygones, cercles, sphres, et tores.

Un paramtre peut tre une constante, ou bien tre un nœud de l'arbre des paramtres i-score et donc provenir d'une source externe : OSC, MIDI, etc. Cela permet l'interactivit : il est possible de mapper des capteurs externes des positions, ou n'importe quel autre paramtre. Il est aussi possible d'utiliser l'coulement du temps l'excution pour faire de l'animation spatiale.

Enfin, on peut dans une certaine mesure dfinir des objets paramtriques :

$$\begin{cases} u = t \\ v = \sin(t) \end{cases} \quad (5)$$

avec t un paramtre externe, pouvant tre le temps. Cependant dans un tel cas, le processus ne conside qu'un point la fois car le fonctionnement est par contraintes. Une autre approche paramtrique plus adapte aux besoins courants de l'criture spatiale est dcrite en section 4.2.2. Il est important

de noter que la seule quation $v = \sin(t)$ ne suffit pas. En effet, si on suppose qu'on applique la mme transformation qu'en 2, ici, la variable d'espace x est non-contrainte. On a donc $y = k$ avec k une constante un instant donn. Le rsultat sera une zone ayant pour forme une droite horizontale qui variera verticalement selon la fonction sinus. De la mme manire, une zone qui ne contraindrait aucune des dimensions correspond l'intgralit de l'espace.

L'autre implmentation :

$$v = \sin(u) \quad (6)$$

est entirement statique car elle ne dpend pas de paramtres externes.

Ensuite, il est possible de dfinir des calculs de relations entre les diffrentes zones. De la mme manire que pour la dfinition des zones, on opre avec une mthode gnrique qui peut ensuite tre sous-classe pour grer des cas soit trs courants, soit ne pouvant s'exprimer simplement de manire mathmatique, ou enfin pouvant tre optimiss par une implmentation en C++. Par exemple, il est possible d'obtenir l'information de collision entre zones. Pour le cas du pointeur on va simplement valuer les valeurs actuelles des informations de dimension par rapport aux autres zones, ce qui est une opration trs rapide. De mme pour les zones de types connus : il est possible d'exhiber des calculs spcia-liss pour chaque relation entre deux types de zones. Enfin, pour des cas impliquant une zone gnrique, non type, nous pouvons chercher l'existence de solutions au systme compos de l'ensemble de leurs quations. Ceci ncessite cependant un solveur capable de rsoudre des systmes d'inquations. Un outil potentiel pour cette application est *nlopt* [21], qui permet de minimiser des systmes non-linaires.

Une autre approche pour les collisions est d'utiliser les mthodes issues du jeu vido, avec des notions de cube englobant ou sphre englobante. De nombreuses approches plus perfectionnes pour la gestion des collisions sont presentes pour rfrence dans [24]. Des moteurs physiques tels que *Bullet* ou *Open Dynamics Engine* [5] pourront par la suite tre utilis cette fin.

Le rsultat des calculs est ensuite expos dans l'arbre i-score, pour tre rutilis par la suite afin de concevoir d'autres zones, ou bien en tant que sortie applique un autre logiciel ou matriel.

i-score tant un environnement ouvert, il est aussi possible de sortir des contraintes du langage mathmatique pour les calculs. En effet, un processus Javascript est offert : il permet de raliser des oprations complexes chaque tic d'horloge. Il est alors possible de rcuprer des informations d'un processus spatial, de les traiter en Javascript, puis de rutiliser leur rsultat au tic suivant. Les automations sont utilisables de la mme manire. Nanmoins, terme, un modle de calcul permettant du chanage dans un seul tic sera ncessaire.

4.2.2. Automations multi-dimensionnelles

En plus du processus Espace, des possibilit d'criture paramtrique, plus simples et adaptes l'criture de trajectoires, sont aussi presentes.

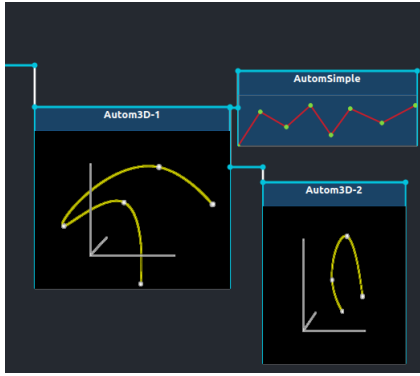


Figure 6. Deux automations 3D et une automation traditionnelle dans i-score. chaque tick d'horloge, la position entre 0 et 1 du temps d'exécution de la contrainte parente sera appliqué la position correspondante dans l'automation.

Il y a plusieurs types d'automation dans i-score :

- Les automations une dimension sont une fonction de transfert du temps t vers un paramètre u . On note $f : t \in [0; 1] \rightarrow D(u)$ ou D est le domaine de définition de la valeur du paramètre ou du service, tel que décrit dans [3].
- Les applications une dimension sont une fonction de transfert entre deux paramètres u, v . On note $f : D(u) \rightarrow D(v)$. Les automations et applications une dimension sont des fonctions définies par partie ; chaque partie peut être une fonction vide, un segment de droite, ou bien la fonction $f : x \mapsto x^k$. Des fonctions supplémentaires peuvent être rajoutées par le biais de plug-ins.
- Les automations deux ou trois dimensions : nous utilisons ici la bibliothèque *vtk* pour afficher et gérer une spline à l'écran, l'aide de l'objet : `vtkSplineWidget`. Ces processus sont visibles en fig. 6

Dans les cas temporels, le temps d'un processus est celui de sa contrainte parente. On a donc pour une paramétrisation sur $t \in [0; 1]$, $t = 0$ au début de la contrainte temporelle, et $t = 1$ la fin de cette contrainte.

Il est possible de choisir entre l'écriture d'un n-uplet de coordonnées, ou bien de plusieurs composantes sur différentes adresses de l'arbre i-score. Cela soulève la question d'un langage de plus haut niveau sur les paramètres, qui rendrait ces choix transparents pour l'utilisateur.

5. IMPLÉMENTATION

La majeure partie de la complexité de l'implémentation provient de la gestion des zones géométriques. Une tentative d'implémentation pour les calculs géométriques sur les zones a été réalisée à l'aide du CAS ² *GiNaC* [4]. Cependant, cela ne s'est pas vraiment adapté : en effet la bibliothèque n'est pas thread-safe ce qui empêche de réaliser des calculs parallèles

sur un autre fil d'exécution. L'implémentation actuelle est donc basée sur le parseur de fonctions mathématiques de la bibliothèque *vtk* [30]. D'autres alternatives ont été envisagées : *SymbolicC++* [33], *GNU Octave* [14], *Sage* ³.

5.1. Sémantique d'exécution du processus Espace

La sémantique générale d'exécution d'i-score est donnée dans [11]. Nous rappelons simplement qu'i-score permet d'ordonner temporellement des processus qui peuvent être définis dans des plug-ins. Lors de la lecture, une horloge globale va demander récursivement tous les processus du scénario qui sont en cours d'exécution quel est leur état actuel. Un état est un ensemble de messages qui peuvent être envoyés par réseau, comme un rappel (*cue*).

Ainsi, s'il y a une réaction, elle se fait au tic d'après. Deux états d'exécution sont présentes en fig. 7 et fig. 8.

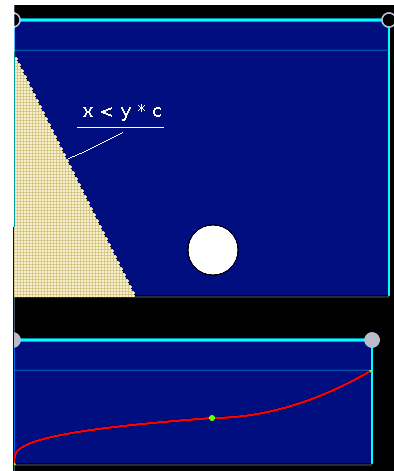


Figure 7. Processus spatial $t = 0s$. La courbe d'automation est appliquée au paramètre c de l'équation de la zone triangulaire.

L'exécution du processus spatial se déroule de la manière suivante :

- Pour chaque zone dont un des paramètres est défini par une adresse externe, récupérer la valeur actuelle de cette adresse.
- Effectuer tous les calculs entre éléments de la zone puis les renvoyer dans l'état.

Dans le cas d'un processus générique, ces calculs peuvent prendre du temps. Ainsi, un mécanisme de requête / réponse sur plusieurs fils d'exécution a été implémenté. Cela permet de ne pas bloquer toute l'exécution du programme ; en revanche le résultat de l'exécution sera décalé de plusieurs tics. Si le processeur est surchargé, les requêtes sont ignorées pour éviter un blocage complet de l'application.

Une autre sémantique est possible : i-score dispose d'un arbre de paramètres interne qu'il est aussi possible de manipuler. Les processus peuvent exposer des paramètres de contrôle cet arbre, qui sont disponibles à l'écriture. i-score est

² . Computer Algebra System, système de calcul formel en français, est un type de logiciel permettant de réaliser du calcul symbolique.

³ . <http://www.sagemath.org/>

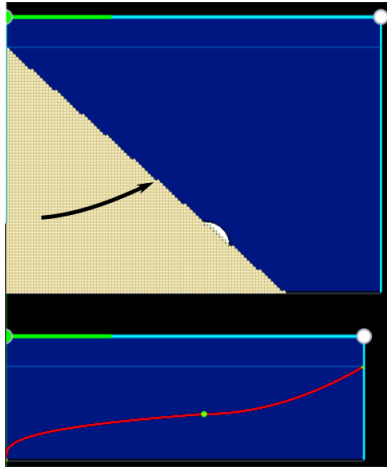


Figure 8. Processus spatial durant l'exécution, $t = 10s$. c a augment, et la zone triangulaire est mise jour en consequence. Il est possible de rcuprer l'information de collision entre le disque et la zone pour dclencher des vnelements ailleurs dans i-score.

donc introspectable depuis son interface graphique. Notamment, le processus spatial expose les valeurs des paramtres prsents pour chacune des zones. Par exemple, pour un disque dont l'equation a t vue en 1, les paramtres x_0, y_0, r sont exposs dans l'arbre interne. De mme, les rsultats des calculs dfinis sont exposs. Il est possible de configurer le processus pour que les calculs enregistrs soient raliss la rception d'un message sur le nœud correspondant, et mis jour dans leurs nœuds respectifs.

Cette question, qui est fondamentalement la diffrence entre un mode *push* et un mode *pull*, se pose aussi pour les autres objets d'i-score. Par exemple, le processus Automation contient une adresse laquelle il crit ses valeurs. Cependant, il est aussi possible de simplement laisser s'excuter la courbe et de permettre d'autres outils d'aller chercher l'information son adresse dans l'arbre interne.

Cela pose la question, plus long terme, d'un graphe de calculs si on dsire avoir des rtroactions et des mises en correspondances complexes. Actuellement, il est possible de raliser des calculs chans la main, en crant des adresses spcifiques et en ordonnant les processus : l'exécution de ces calculs est globale. Mais par la suite, un modele de calcul plus complet sera ncessaire pour rendre manifeste les liens qui peuvent exister entre diffrents processus, la manire d'environnements tels que OpenMusic [7], Max/MSP ou PureData.

5.2. Rendu

On affecte chacune des dimensions de l'espace une dimension fixe de la zone graphique : x, y pour l'instant avec pour objectif du fonctionnement en 3D.

L'affichage se fait de manire specialise pour les objets connus fix l'avance, comme cercle, pointeur, Cela permet d'utiliser les primitives de la bibliothque Qt des

fins d'optimisation lors de la phase d'dition. Pour les objets gnriques, on ralise pour l'instant une sur-pixellisation, en valuant la formule pour plusieurs points et en plaant un rectangle si le point vrifie toutes les contraintes.

Comme c'est une opration lourde, chaque zone de ce type effectue le rendu dans un fil d'exécution spar. Lorsqu'un calcul termine, les rectangles ainsi calculs sont envoys la bibliothque d'affichage.

C'est important pour avoir un rendu fluide l'exécution. Le nombre d'images par seconde du processus spatial sera potentiellement faible, mais il n'y aura pas de blocage de l'interface graphique pendant le rendu.

D'autres mthodes sont tudier, par exemple en utilisant des algorithmes de triangulation qui ont t voques plus tt. Cela aurait l'avantage de produire un format de donnes que les cartes graphiques peuvent consommer beaucoup plus aisment.

Enfin, la question de l'affichage pour des dimensions non-spatiales, telles que le temps ou une chelle de couleur, reste ouverte.

5.3. dition

La cration d'objets se fait actuellement par un panneau de configuration. L'dition va modifier pour les formes connues des paramtres dfinis.

Les oprations de base sur les zones sont dplacement, rotation et mise l'chelle.

On applique les transformations aux variables d'espace de l'objet.

- La translation et la mise l'chelle sont de simples transformations affines.
- La rotation par un angle θ dans le plan s'exprime par le changement de coordonnes suivant en dimension deux :

$$\begin{cases} x' = x \cos \theta - y \sin \theta \\ y' = x \sin \theta + y \cos \theta \end{cases}$$

Une formule gnrale de la rotation en n dimensions est donne dans [1] et pourra tre applique par la suite.

On peut ensuite contrler ces transformations la souris, l'aide d'outils, comme le font la majorit des logiciels d'images de synthse et de jeux vido, ou bien y avoir accs depuis l'arbre de paramtres.

6. CONCLUSION

L'criture spatiale dans un cadre scnographique ou musical prsente de nombreux choix en terme de modlisation, d'implmentation, et de representation. Nous avons prsent un tat de l'art des outils d'criture spatiale dans ces domaines, dont une version plus complte est prsente dans le rapport du projet de recherche OSSIA. Par la suite, les diffrentes mthodes pour aboutir une criture spatiale prcise sont prsentes et discutes. Le choix d'implmentation pour le logiciel i-score se porte sur deux modes : le premier est un mode mixte orient gomtrie, permettant des dfinitions

de zones spatiales gnriques ainsi que spcialises, dont l'intrt repose dans sa capacit reprsenter une scne spatiale dont on peut extraire des informations sur les relations entre les objets qui la composent. Le second est un modle paramtrique traditionnel adapt l'criture de trajectoires. L'utilisation d'objets gnriques n'offre actuellement pas des performances optimales. Cependant, le fait de travailler dans le domaine mathmatique offre terme les plus grandes possibilitis pour des amliorations et optimisations. Ces deux modles sont utilisss dans des projets de scnographie interactives mlant du son ainsi que d'autres medias.

Ces modles sont tous deux adapts un travail dans des espaces cartsiens, sur des paramtres linaires. Cependant, un travail reste faire sur un outil permettant de traiter des paramtres ou espaces de paramtres diffrents, comme les espaces colorimtriques. Cela implique un travail sur un ty-page plus fort des paramtres manipulss dans i-score. Une dynamicit de l'criture pourra aussi tre intressante : par exemple, crer des zones lors de l'excution pourrait offrir de nouvelles perspectives d'criture.

Un travail sur l'extension en trois dimensions des splines, les surfaces NURBS, pourra aussi tendre les possibilitis de la mthode, notamment en permettant une intgration avec des logiciels ddiss la cration de contenus 3D.

7. REMERCIEMENTS

Les auteurs tiennent remercier les lves ayant ralis le logiciel de simulation pour Metabots : Maxime Paillassa et Akan Levy. Ces travaux sont issus du projet de recherche OSSIA, financ par l'Agence Nationale de la Recherche sous la rfrence ANR-12-COORD-0024.

8. REFERENCES

- [1] Antonio Aguilera and Ricardo Pérez-Aguila. General n-dimensional rotations. In *12th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*. UNION Agency - Science Press, Plzen, Czech Republic, 2004.
- [2] James F Allen. Towards a general theory of action and time. *Artificial intelligence*, 23(2) :123–154, 1984.
- [3] Pascal Baltazar, Tho de la Hogue, and Myriam Desainte-Catherine. i-score, an interactive sequencer for the intermedia arts. In *Proceedings of the ICMC - SMC 2014 Joint Conference*, 2014.
- [4] Christian Bauer, Alexander Frink, and Richard Kreckel. Introduction to the ginac framework for symbolic computation within the c++ programming language. *Journal of Symbolic Computation*, 33(1) :1–12, 2002.
- [5] Adrian Boeing and Thomas Bräunl. Evaluation of real-time physics simulation systems. In *Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia*, pages 281–288. ACM, 2007.
- [6] Jean-Daniel Boissonnat, Olivier Devillers, Monique Teillaud, and Mariette Yvinec. Triangulations in cgal. In *Proceedings of the sixteenth annual symposium on Computational geometry*, pages 11–18. ACM, 2000.
- [7] Jean Bresson, Carlos Agon, and Gérard Assayag. Openmusic : Visual programming environment for music composition, analysis and research. In *Proceedings of the 19th ACM International Conference on Multimedia*, MM '11, pages 743–746, New York, NY, USA, 2011. ACM.
- [8] Emil Brink, Eskil Steenberg, and Gert Svensson. The verse networked 3d graphics platform. In *Linköping Electronic Conference Proceedings. SIGRAD Conference*, pages 44–48. Citeseer, 2006.
- [9] Benjamin Cabaud and Laurent Pottier. Le contrôle de la spatialisation multi-sources - nouvelles fonctionnalités dans holophon version 2.2. *Actes des Journées d'Informatique Musicale*, 2002.
- [10] Dan Casas, Margara Tejera, J. Guillemaut, and Adrian Hilton. 4d parametric motion graphs for interactive animation. 19(5) :762–773, 2013.
- [11] Jean-Michel Celerier, Pascal Baltazar, Clment Boscut, Nicolas Vuaille, Jean-Michel Couturier, and Myriam Desainte-Catherine. Ossia : Towards a unified interface for scoring time and interaction. 2015.
- [12] Tho De La Hogue, Julien Rabin, and Laurent Garnier. Jamoma modular : une librairie c++ ddiee au dveloppement d'applications modulaires pour la cration. *Proceedings of the 17es Journées d'Informatique Musicale, Saint-Etienne, France*, 2011.
- [13] Olivier Delerue. Spatialisation du son et programmation par contraintes : le systme MusicSpace, 2004.
- [14] John Wesley Eaton, David Bateman, and Søren Hauberg. *Gnu Octave*. Network theory, 1997.
- [15] Emile Ellberger, Germn Toro Perez, Johannes Schuett, Giorgio Zoia, and Linda Cavaliero. Spatialization symbolic music notation at ICST. 2014.
- [16] Dominique Fober, Jean Bresson, Pierre Couprie, and Yann Geslin. Les nouveaux espaces de la notation musicale. In *Journes d'Informatique Musicale*.
- [17] Alfonso Gerevini and Bernhard Nebel. Qualitative spatio-temporal reasoning with RCC-8 and allen's interval calculus : Computational complexity. In *ECAI*, volume 2, pages 312–316.
- [18] Tovi Grossman, Ravin Balakrishnan, and 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*, pages 185–192. ACM.
- [19] Garin Hiebert. Openal 1.1 specification and reference. 00010.

- [20] Guillaume Jacquemin, Thierry Coduys, and Matthieu Ranc. Iannix 0.8. pages 107–15.
- [21] Steven G Johnson. The nlopt nonlinear-optimization package, 2014.
- [22] M Kaltenbranner, Sergi Jorda, Gunter Geiger, and Marcos Alonso. The reactable : A collaborative musical instrument. In *Enabling Technologies : Infrastructure for Collaborative Enterprises, 2006. WE-TICE'06. 15th IEEE International Workshops on*, pages 406–411. IEEE.
- [23] Mika Kuuskankare and Mikael Laurson. Expressive notation package. 30(4) :67–79.
- [24] Ming Lin and Stefan Gottschalk. Collision detection between geometric models : A survey.
- [25] Bertrand Merlier. *Vocabulaire de l'Espace en Musiques lectroacoustiques*.
- [26] Luiz Naveda and Ivani Santana. "topos" toolkit for pure data : exploring the spatial features of dance gestures for interactive musical applications.
- [27] L Nonato, Rosane Minghim, MCF Oliveira, and Geovan Tavares. A novel approach for delaunay 3d reconstruction with a comparative analysis in the light of applications. In *Computer Graphics Forum*, volume 20, pages 161–174. Wiley Online Library, 2001.
- [28] Natanael Olaiz, Pau Arumi, Toni Mateos, and David Garcia. *3D Audio with CLAM and Blender ?s Game Engine*. na, 2009.
- [29] Yuya Sasamoto, Michael Cohen, and Julian Vilegas. Controlling spatial sound with table-top interface. In *Awareness Science and Technology and Ubi-Media Computing (iCAST-UMEDIA), 2013 International Joint Conference on*, pages 713–718. IEEE.
- [30] Will J Schroeder, Bill Lorensen, and Ken Martin. *The Visualization Toolkit*. Kitware, 2004.
- [31] James Sheridan, Gaurav Sood, Thomas Jacob, Henry Gardner, and Stephen Barrass. Soundstudio 4d : A VR interface for gestural composition of spatial soundscapes. In *ICAD*.
- [32] Paul S Strauss and Rikk Carey. An object-oriented 3d graphics toolkit. In *ACM SIGGRAPH Computer Graphics*, volume 26, pages 341–349. ACM, 1992.
- [33] Kiat Shi Tan, Willi-Hans Steeb, and Yorick Hardy. *SymbolicC++ : An Introduction to Computer Algebra using Object-Oriented Programming*. Springer Science & Business Media, 2012.
- [34] Graham Wakefield and Wesley Smith. *Cosm : A toolkit for composing immersive audio-visual worlds of agency and autonomy*. Ann Arbor, MI : MPubli-shing, University of Michigan Library. 00007.