



**HAL**  
open science

## Représentation des données de contrôle pour la spatialisation dans OpenMusic

Jean Bresson, Carlos Agon, Marlon Schumacher

► **To cite this version:**

Jean Bresson, Carlos Agon, Marlon Schumacher. Représentation des données de contrôle pour la spatialisation dans OpenMusic. Journées d'Informatique Musicale, May 2010, Rennes, France. hal-01157046

**HAL Id: hal-01157046**

**<https://hal.science/hal-01157046>**

Submitted on 27 May 2015

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

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

# REPRÉSENTATION DES DONNÉES DE CONTRÔLE POUR LA SPATIALISATION DANS OPENMUSIC

*Jean Bresson, Carlos Agon*  
IRCAM - CNRS UMR STMS  
{jean.bresson,carlos.agon}@ircam.fr

*Marlon Schumacher*  
IDMIL - DCS - CIRMMT  
Schulich School of Music of McGill University  
marlon.schumacher@music.mcgill.ca

## RÉSUMÉ

La prise en compte de l'espace dans les démarches musicales contemporaines entraîne une nouvelle réflexion sur les méthodes et environnements informatiques de composition. Nous présentons des travaux actuels réalisés dans l'environnement de composition assistée par ordinateur OpenMusic. Plusieurs directions sont explorées, concernant la génération de trajectoires dans l'espace, les structures de contrôle pour la localisation et la spatialisation des sources sonores, ou encore la création de sons spatialisés au sein de cet environnement. L'export et la communication de données avec des systèmes de spatialisation externes sont également envisagés, pour lesquels nous proposons une solution utilisant le format SDIF.

## 1. INTRODUCTION

Malgré les progrès réalisés dans les techniques de rendu sonore spatialisé, la prise en compte de l'espace dans les processus de création musicale reste un aspect problématique et relativement peu exploré en composition assistée par ordinateur.

La majorité des outils existant actuellement se situe dans le domaine du traitement audio temps-réel ou dans des environnements et stations audio-numériques, et propose des interfaces de contrôle permettant de déterminer de façon plus ou moins directe un "panning" d'amplitude des sources sonores sur différentes configurations de haut-parleurs.<sup>1</sup> La détermination des paramètres spatiaux (trajectoires, effets de salles, etc.) reste cependant relativement limitée par un contrôle généralement décorrélé de la perception, mais également des autres paramètres musicaux. Ces limitations, par ailleurs, s'affirment d'autant plus lorsque le nombre de sources sonores est élevé et/ou lorsque les configurations de restitution ne sont pas fixées dès le départ.

Le Spatialisateur de l'IRCAM [13] est un exemple d'application intégrée dans l'environnement Max/MSP, permettant de contrôler les paramètres de spatialisation à l'aide de données telles que les positions et mouvements de sources dans l'espace, les caractéristiques acoustiques des salles ou autres paramètres perceptifs. Cependant, ces

paramètres restent généralement déconnectés de l'univers compositionnel et des représentations symboliques qui permettraient de les insérer dans une démarche musicale.

La recherche de contrôle musical de plus haut niveau pour la spatialisation a été abordée dans différents projets tels que le logiciel MusicSpace [10] qui permet de contrôler les positions de différentes sources sonores à l'aide d'un système de contraintes, établies et propagées dynamiquement entre celles-ci. L'interface Holo-Edit, créée dans le cadre du projet Holophon [8] propose une autre approche et permet de créer et éditer des trajectoires tridimensionnelles graphiquement ou à l'aide de fonctions de génération et de transformation [19]. Cet aspect fonctionnel, ou algorithmique, également présent dans des projets plus anciens tels que MoveInSpace [23], est un pas significatif dans l'idée d'un contrôle compositionnel des paramètres spatiaux. En effet il est important pour les compositeurs voulant intégrer l'espace dans leurs démarches musicales de pouvoir définir leurs propres processus génératifs et de les lier à un univers ou un projet compositionnel donné [2].

Le contrôle de la spatialisation dans le contexte de la composition assistée par ordinateur (CAO) a également fait l'objet de précédentes expériences dans l'environnement OpenMusic [3]. Avec OpenSpace [11], il a été envisagé de contrôler le système de contraintes de MusicSpace, ajoutant et modifiant ces contraintes dynamiquement dans un déroulement temporel déterminé depuis l'environnement de CAO. Un autre projet important, à la base de nos travaux actuels, est la bibliothèque OMSpat [17] qui permettait d'associer trajectoires et paramètres spatiaux générés dans OpenMusic à des sources sonores, par l'intermédiaire d'une structure matricielle formatée et transférée à une interface de contrôle pour le Spatialisateur.

Les travaux que nous présentons étendent le système OMSpat sous différents points de vue, reprenant et généralisant l'idée des objets et interfaces primitives créées alors pour la représentation des courbes et trajectoires dans l'espace, ainsi que celle d'une représentation matricielle pour la description des scènes sonores. Rattachée au système de contrôle de la synthèse OMChroma [1], cette représentation matricielle peut générer un rendu sonore spatialisé directement dans OpenMusic. Nous insisterons ici sur une solution complémentaire consistant à transférer

<sup>1</sup>. Une revue et description des principaux outils et logiciels de spatialisation est disponible sur le site  
[http://multiphonie.free.fr/log\\_spat.htm](http://multiphonie.free.fr/log_spat.htm)

ces données vers des systèmes externes de spatialisation, en utilisant le format SDIF.

## 2. GÉNÉRATION DE TRAJECTOIRES

Une spécificité du contrôle de spatialisation dans le contexte de la CAO est de travailler sur des représentations abstraites des objets musicaux, et non sur des données de paramétrage comme dans le cas des contrôleurs audio-numériques. Nous nous intéresserons principalement ici aux trajectoires tridimensionnelles, regroupant des points localisés dans l'espace et leurs évolutions temporelles. Le même type de représentations pourra cependant être utilisé pour des ensembles de positions statiques, ou même pour d'autres données telles que la directivité des sources sonores.

De nouveaux objets viennent compléter les représentations traditionnelles des courbes dans OpenMusic. L'objet *3DC* est une suite de points en 3 dimensions, et l'objet *3D-Trajectory* étend ce dernier à la dimension temporelle, en permettant d'associer une date aux différents points (tous ou certains points d'intérêt). Les outils de traitement et de génération de courbes disponibles dans OpenMusic (par exemple le rééchantillonnage ou la génération de courbes B-splines) permettent de créer ces données parfois complexes à partir de nombres réduits de points entre lesquels une trajectoire peut être calculée suivant une résolution donnée (spatiale ou temporelle).

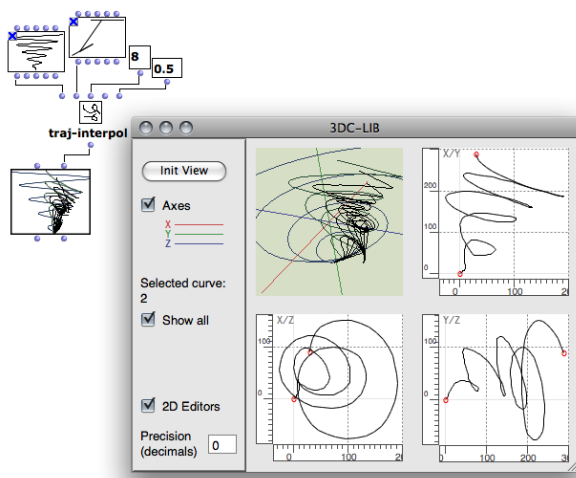


Figure 1. Création et édition de courbes et trajectoires 3D.

Suivant un principe de dualité entre processus et objets musicaux, courbes et trajectoires peuvent être générées par des processus fonctionnels et algorithmiques au sein de l'environnement de programmation visuelle, mais également directement à l'aide d'éditeurs graphiques permettant de dessiner les courbes, placer / déplacer les points, modifier les points d'inflexion ou les repères temporels. La figure 1 montre un objet *3DC-Lib* (collection d'objets *3DC*) créé à l'aide d'une fonction d'interpolation

dans un programme visuel, ainsi que l'éditeur graphique associé à cet objet.

## 3. MATRICES ET REPRÉSENTATION DES SCÈNES SONORES SPATIALISÉES

Comme nous l'avons évoqué en introduction, les structures matricielles, utilisées notamment pour la synthèse sonore dans OpenMusic [1], présentent un intérêt pour le contrôle de la spatialisation et la représentation des scènes sonores spatialisées sous forme d'objets compacts, symboliques et opérationnels.

Dans une de ses dimensions, une matrice permet en effet de mettre en relation une source sonore et ses différents paramètres de spatialisation (positions, trajectoires, mais éventuellement aussi d'autres paramètres acoustiques comme la réverbération, la directivité, etc.) Dans sa deuxième dimension, elle met en relation les différents éléments constituant un son (dans le cas de la synthèse sonore) ou une scène sonore spatialisée, associant entre elles les différentes valeurs de chaque paramètre du processus. A titre d'exemple, il est ainsi possible d'associer aux différents composants d'une scène spatialisée une même trajectoire, d'établir des rapports entre leurs trajectoires respectives, ou encore de définir un processus qui générera dynamiquement l'ensemble de ces trajectoires.

La figure 2 montre un exemple de construction d'une "matrice de spatialisation", associant sources sonores et paramètres de spatialisation. Cette structure constitue une représentation abstraite susceptible d'être intégrée dans un processus musical de plus haut niveau, et éventuellement transférée vers un système de spatialisation sonore.

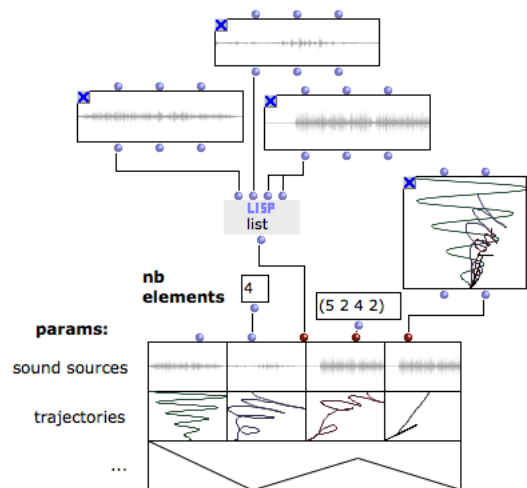


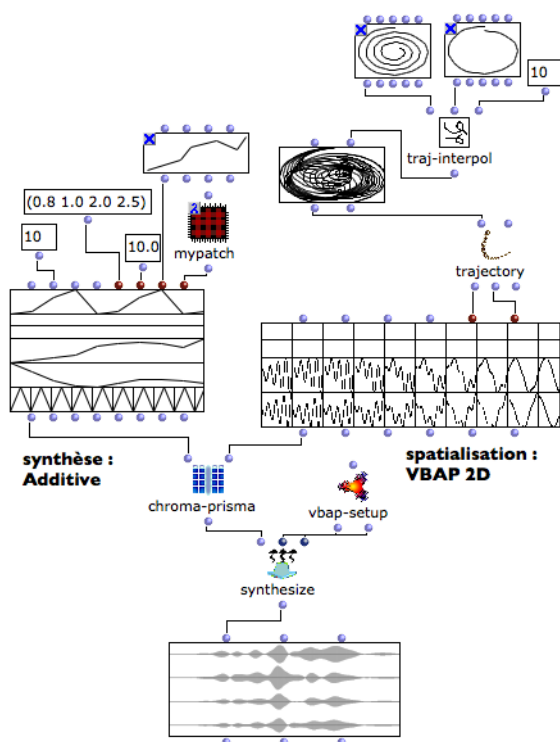
Figure 2. Objets et sources sonores spatialisées dans OpenMusic : représentation matricielle.

Ce transfert pourra avoir lieu dans OpenMusic, comme illustré dans la section 4, ou par le formatage et la communication avec des environnements externes (section 5).

#### 4. SYNTHÈSE ET RENDU SONORES SPATIALISÉS DANS OPENMUSIC : OMPRISMA

OMPrisma [21] est une bibliothèque OpenMusic étendant à la spatialisation le système de contrôle de la synthèse OMChroma. Ce système propose un ensemble de classes correspondant à différents processus de synthèse (ou de spatialisation, avec OMPrisma). Une classe dans OMChroma (et OMPrisma) est une matrice associée à une unité de génération ou traitement de signal (implémentée sous forme de code ou *instrument* Csound [5]). Ainsi les différents paramètres de cette unité de traitement (correspondant au *score* dans Csound) sont générés à partir des valeurs de la matrice suivant l'une de ses dimensions, et instanciés un nombre de fois déterminé par sa deuxième dimension. Un appel au compilateur Csound permet ensuite de générer un fichier audio à partir de l'instrument associé et le *score* généré par la matrice.

Suivant ce principe, OMPrisma propose des modules de spatialisation de sources sonores basés sur des techniques telles que le panning sur différents canaux, le VBAP (*vector-based amplitude panning* [20]), ou encore la synthèse *ambisonics* [9]. Celles-ci permettent de générer des scènes sonores complexes en relation aux processus compositionnels développés dans OM. L'interchangeabilité des différentes classes permet de concevoir ces processus et scènes sonores de façon relativement indépendante du système de rendu et de reproduction spatialisée.



**Figure 3.** Synthèse sonore spatialisée avec OMChroma et OMPrisma : chaque composant du processus de synthèse (partiels, ici dans le cas d'une synthèse additive) est spatialisé suivant une trajectoire calculée par interpolation.

Par ailleurs, il est possible aussi de fusionner les classes de spatialisation de OMPrisma avec les différentes classes proposées pour la synthèse sonore dans OMChroma (par exemple, synthèse additive, FM, FOF, granulaire, etc.) Chaque composant d'un processus de synthèse (respectivement, chaque partiel, instanciation d'un générateur FM, fonction d'un banc de FOF, grain sonore, etc.) peut alors être associé à un ensemble de paramètres et à un module de traitement intégré pour sa spatialisation (voir figure 3). Couplé à l'environnement de contrôle, ce système implémente une approche générique de la notion de synthèse sonore spatialisée (également abordée dans des travaux tels que [24] ou [15]) adaptée au contexte compositionnel.

#### 5. EXPORT DES DONNÉES : UNE PROPOSITION AVEC LE FORMAT SDIF

La standardisation des formats d'échange pour la spatialisation est une question centrale, dans la mesure où un nombre croissant de technologies se développent, et où les activités de contrôle, de rendu et de reproduction sont très souvent réalisées dans des environnements et configurations variables. Au delà des possibilités offertes par la spatialisation réalisée dans OpenMusic (via OMPrisma et Csound), il serait avantageux, en effet, de générer des données également exploitables dans d'autres environnements, notamment dans les environnements de traitement temps-réel comme le Spatialisateur ou avec des systèmes plus complexes comme la WFS [4].

Différentes tentatives de standardisation ont été proposées pour la description de scènes sonores spatialisées, notamment autour des formats MPEG ou du standard XML (voir le panel dédié à ce sujet lors de la conférence ICMC 2008 [14]). Parmi celles-ci, les formats de stockage (par exemple ASDF [12], pour reprendre l'une des propositions du précédent panel, mais également des formats plus généraux comme VRML/X3D) sont généralement axés sur des descriptions et spécifications de haut niveau (par exemple avec des déclarations d'objets et de transformations telles que des rotations ou translations).

Dans le cadre qui nous concerne ici, cependant, nous souhaitons décrire des scènes sonores d'un point de vue "dénotationnel", c'est-à-dire écrire des fichiers décrivant des trajectoires et courbes de paramètres échantillonnés et dans leur résolution finale (cet échantillonnage pouvant d'ailleurs être pris en charge par l'environnement de contrôle en tant que spécification "compositionnelle"). Ces descriptions pourront alors être lues et interprétées directement, sans déduction ni calcul, par différents environnements de rendu spatialisé.

A l'inverse des précédents, les formats ou protocoles dédiés à la communication temps-réel (par exemple SpatDIF [18]) proposent des descriptions plus directes et précises mais n'offrent actuellement pas de solution de stockage satisfaisante.<sup>2</sup>

<sup>2</sup>. Dans l'idéal, un standard devrait à terme inclure ces deux modalités et permettre à la fois le transfert (*streaming*) et le stockage des données, à l'instar du protocole MIDI et du Standard MIDI File [16].

## 5.1. Le format : SDIF

Suivant les précédentes évolutions de l’environnement de CAO en direction du domaine de l’analyse et la synthèse sonore, nous avons choisi d’utiliser le format SDIF (*Sound Description Interchange Format* [22]) comme support de nos descriptions spatiales dans OpenMusic.<sup>3</sup> Plusieurs raisons peuvent être évoquées à cela, parmi lesquelles :

- L’existence d’outils permettant la manipulation et le traitement des données SDIF dans OpenMusic, et l’utilisation préalable et établie de ce format pour le stockage et transfert des descriptions sonores [6] ;
- L’extensibilité et l’ouverture du format, qui propose un protocole aisé pour l’ajout de nouveaux types de données ;
- La structure du format, qui permet d’envisager les données sous forme de flux temporels multiples.

Le format SDIF se base sur une description matricielle (qui convient donc a priori à notre représentation interne, voir section 3) décrivant les valeur d’un certain nombre de paramètres (ou champs de description – *fields*) pour un certain nombre de composants. Ces matrices ont un “type” (identifié par une *signature* de 4 caractères) et sont incluses dans des *frames* déterminant un ordonnancement temporel. Les *frames* ont également une signature correspondant à un type, et un identifiant de flux (*stream ID*) permettant de décrire et différencier un ensemble de flux entrelacés dans un même fichier.

Il existe un certain nombre types SDIF standards et pré-définis, correspondant aux descriptions sonores les plus courantes, mais il est possible d’étendre ceux-ci par de nouveaux paramètres, ou d’en définir de nouveaux. La déclaration de types peut avoir lieu au niveau global dans la librairie SDIF, pour les types standardisés, ou localement à l’intérieur d’un fichier dans les autres cas. Une fois ces types déclarés, une application lisant les données dans un fichier est ainsi capable, d’après le type des matrices, de les identifier et éventuellement de les interpréter de façon adéquate.

## 5.2. Déclaration des types

Notre description, dans un premier temps, comprend uniquement les trajectoires tridimensionnelles des sources sonores dans l’espace. Voici la déclaration du type de matrice proposé selon le protocole de spécification SDIF :

```
1MTD XPOS {x, y, z}
```

Nous définissons ici le type de matrice "XPOS" (MTD = *Matrix Type Definition*) dont les trois champs sont les coordonnées cartésiennes des points dans l’espace.<sup>4</sup>

3. SDIF a également été envisagé comme support de stockage pour des données de contrôle SpatDIF [18].

4. Selon les conventions, les types SDIF standardisés commencent généralement par un numéro – "1" la plupart du temps, et les types plus expérimentaux par un "X".

D’autres types peuvent être déclarés sur ce modèle, correspondant par exemple aux coordonnées sphériques ou cylindrique :

```
1MTD XSPH {azimuth, zenith, distance}
1MTD XCYL {azimuth, distance, elevation}
```

Comme nous l’avons dit, les matrices sont intégrées dans un flux temporel par l’intermédiaire de *frames*, pour lesquels il est nécessaire de définir également un type. Le type "XSPA" défini ci-dessous (FTD = *Frame Type Definition*) correspond à un *frame* SDIF pouvant contenir des matrices de type XPOS, XSPH, ou XCYL.

```
1FTD XSPA {
  XPOS cartesian_coord;
  XSPH spherical_coord;
  XCYL cylindrical_coord;
}
```

La figure 4 montre un exemple de stream SDIF (séquence de *frames*) décrivant l’évolution de la position d’une source sonore dans l’espace.

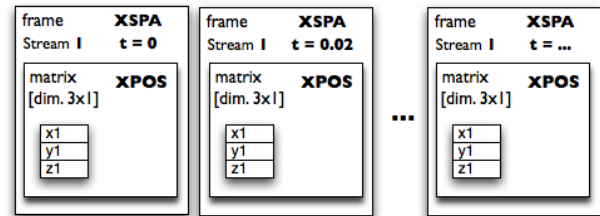


Figure 4. Séquence de *frames* SDIF de type "XSPA" décrivant l’évolution dans l’espace d’une source sonore.

De la même façon peut être envisagée la définition de types de matrices correspondant aux autres paramètres entrant en jeu dans la spatialisation, comme les effets de salle ou la directivité des sources. Concernant les effets de salles, nous pouvons par exemple imaginer un nouveau type de matrice (par exemple "XRFX" pour “room effects”) qui serait inclus dans un flux de *frames* d’un nouveau type (par exemple lui aussi de signature "XRFX") :

```
1MTD XRFX {param1, param2, ...}
1FTD XRFX { XRFX room_effects; }
```

Si l’on souhaite attacher ces paramètres à des sources particulières, il est également possible d’étendre le type de *frame* "XSPA" pour y autoriser les matrices de type "XRFX" :

```
1FTD XSPA {
  XPOS cartesian_coord;
  XSPH spherical_coord;
  XCYL cylindrical_coord;
  XRFX room_effects;
}
```

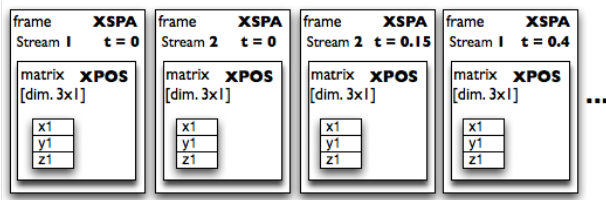
De cette manière, les paramètres de salle, soit au niveau général, soit au niveau de chaque source individuelle, peuvent être inclus sans incidence dans les flux de contrôle SDIF au(x) moment(s) où ces paramètres auront à être communiqués ou modifiés.

### 5.3. Description de scènes sonores “polyphoniques”

Les aspects polyphoniques, soit la possible multiplicité des sources sonores présentes dans une scène spatialisée, peuvent être considérés de différentes manières.

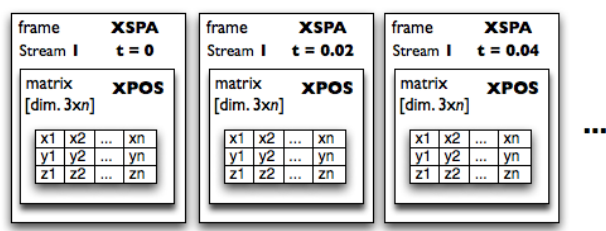
L’une d’elles serait une description mettant en jeu un certain nombre sources sonores identifiées (typiquement, des fichiers audio) : dans ce cas il est préférable de générer un flux séparé pour chaque source, ce qui permettra de leur assigner des granularités ou précisions de contrôle indépendantes (par exemple certaines peuvent être immobiles et décrites par un ou peu de *frames*, quand d’autres décriront des mouvements nécessitant une plus grande précision temporelle). La figure 5 illustre ce type de description où chaque source est liée à un *stream* SDIF et donc encapsulée dans un *frame* indépendant.

Le système de spécification de relations entre les *streams* proposé dans [7] nous invite à ce type d’organisation dans laquelle les flux peuvent être liés les uns aux autres ou à des ressources externes par l’intermédiaire de tables déclarées en tête de fichier.



**Figure 5.** Séquence de *frames* SDIF “XSPA” décrivant l’évolution dans l’espace de deux sources sonores, correspondant respectivement aux *streams* 1 et 2.

Lorsque les sources ne sont pas des ressources identifiées mais par exemple les multiples éléments (grains, partiels, etc.) d’un processus de synthèse sonore (comme dans le cas de la synthèse sonore spatialisée, voir section 4), il peut être préférable d’envisager un flux unique de *frames* dont les matrices n’auront pas un, comme dans le cas précédent, mais un ensemble d’éléments synchronisés. Chacun de ces éléments aura alors des caractéristiques spatiales individuelles mais spécifiées dans une même structure localisée dans le temps. La figure 6 illustre ce deuxième cas.



**Figure 6.** Séquence de *frames* SDIF “XSPA” décrivant l’évolution dans l’espace de  $n$  sources sonores. Chaque *frame* décrit les  $n$  positions dans une même matrice.

Enfin les deux cas envisagés précédemment peuvent être mis en œuvre simultanément, permettant ainsi de grouper certains éléments dans un flux commun et en parallèle à un certain nombre d’autres flux indépendants.

La figure 7 montre un exemple de description de trajectoires associées à des sources sonores, générée comme “matrice de spatialisation” dans OM et écrite sous forme de fichier SDIF suivant les précédentes spécifications. L’éditeur SDIF-Edit [6] est utilisé pour la visualisation de ces trajectoires.

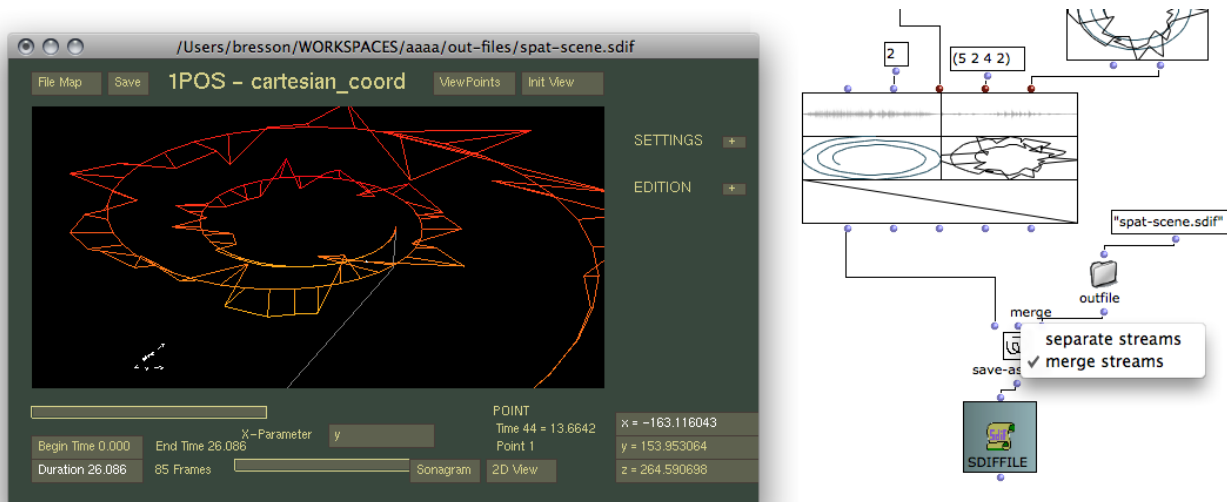
## 6. CONCLUSION

L’accessibilité des techniques de spatialisation sonore et l’interpénétration de plus en plus courante des pratiques musicales “symboliques” et électroacoustiques font de la spatialisation un enjeu important des recherches en composition assistée par ordinateur.

Nous avons essayé de donner un aperçu de nos travaux actuels sur ce sujet, et des orientations choisies dans l’environnement OpenMusic. Les outils que nous proposons visent avant tout une ouverture et une généralité qui doivent leur permettre de s’adapter à la multiplicité des approches et aux diversités esthétique et opératoire des compositeurs. En particulier, nous avons mis en avant des choix de représentations et de formats dans l’objectif d’insérer notre environnement dans une interaction possible, et à différents niveaux des processus compositionnels, avec les différents outils de création musicale existant à l’heure actuelle.

## 7. REFERENCES

- [1] C. Agon, M. Stroppa, and G. Assayag. High Level Control of Sound Synthesis in OpenMusic. *Proc. International Computer Music Conference*, Berlin, Germany, 2000.
- [2] G. Assayag. Computer assisted composition today. *Ist symposium on music and computers*, Corfu, 1998.
- [3] G. Assayag, C. Rueda, M. Laurson, C. Agon and O. Delerue. Computer Assisted Composition at IR-CAM : From PatchWork to OpenMusic. *Computer Music Journal*, 23(3), 1999.
- [4] A. J. Berkhout, D. de Vries, and P. Vogel. Acoustic control by wave field synthesis. *Journal of the Acoustical Society of America*, 93, 1993.
- [5] R. Boulanger, editor. *The Csound Book. Perspectives in Software Synthesis, Sound Design, Signal Processing and Programming*. MIT Press, 2000.
- [6] J. Bresson and C. Agon. SDIF Sound Description Data Representation and Manipulation in Computer-Assisted Composition. *Proc. International Computer Music Conference*, Miami, USA, 2004.
- [7] J. J. Burred, C. E. Cella, G. Peeters, A. Röbel, and D. Schwartz. Using the SDIF Sound Description Interchange Format for Audio Features. *Proc. Inter-*



**Figure 7.** Sauvegarde de trajectoires sous forme de fichier SDIF. Dans cet exemple les trajectoires sont contenues dans un même *stream* SDIF (option “merge streams”) et donc visualisées simultanément. La visualisation courante sur la fenêtre SDIF-Edit (à gauche), montre l’évolution des valeurs  $x$  par rapport à  $y$  en fonction du temps, pour les deux trajectoires.

*national Conference on Music Information Retrieval*, Philadelphia, USA, 2008.

[8] B. Cabaud and L. Pottier. Le contrôle de la spatialisation multi-sources – Nouvelles fonctionnalités dans Holophon version 2.2. *Actes des Journées d’Informatique Musicale*, Marseille, France, 2002.

[9] J. Daniel. *Représentation de champs acoustiques, applications à la transmission et à la reproduction de scènes sonores complexes dans un contexte multimedia*. PhD thesis, Université Pierre et Marie Curie - Paris 6, France, 2001.

[10] O. Delerue. *Spatialisation du son et programmation par contraintes : le système MusicSpace*. PhD thesis, Université Pierre et Marie Curie - Paris 6, France, 2004.

[11] O. Delerue and C. Agon. OpenMusic + MusicSpace = OpenSpace. *Actes des Journées d’Informatique Musicale*, Issy-les-moulineaux, France, 1999.

[12] M. Geier and S. Spors. ASDF : Audio Scene Description Format. *Proc. International Computer Music Conference*, Belfast, Ireland, 2008.

[13] J.-M. Jot and O. Warusfel. A Real-Time Spatial Sound Processor for Music and Virtual Reality Applications. *Proc. International Computer Music Conference*, Banff, Canada, 1995.

[14] G. S. Kendall, N. Peters, and M. Geier. Towards an Interchange Format for Spatial Audio Scenes. *Proc. International Computer Music Conference*, Belfast, Ireland, 2008.

[15] D. Kim-Boyle. Spectral Spatialization - an Overview. *Proc. International Computer Music Conference*, Belfast, Ireland, 2008.

[16] MIDI Manufacturers Association, *Complete MIDI 1.0 Detailed Specification*, 1996.

[17] G. Nouno and C. Agon. Contrôle de la spatialisation comme paramètre musical. *Actes des Journées d’Informatique Musicale*, Marseille, France, 2002.

[18] N. Peters, S. Ferguson, and S. McAdams. Towards a Spatial Sound Description Interchange Format (Spat-DIF). *Canadian Acoustics*, 35(3), 2007.

[19] L. Pottier. Dynamical Spatialisation of sound. HOLOPHON : a graphical and algorithmical editor for  $\Sigma 1$ . *Proc. International Conference on Digital Audio Effects (DAFx-98)*, Barcelona, Spain, 1998.

[20] V. Pulkki. Virtual sound source positioning using vector base amplitude panning. *Journal of the Audio Engineering Society*, 45(6), 1997.

[21] M. Schumacher and J. Bresson. Compositional Control of Periphonic Sound Spatialization. *Second International Symposium on Ambisonics and Spherical Acoustics*, Paris, France, 2010.

[22] D. Schwartz and M. Wright. Extensions and Applications of the SDIF Sound Description Interchange Format. *Proc. International Computer Music Conference*, Berlin, Germany, 2000.

[23] T. Todoroff, C. Traube, and J.-M. Ledent. NeXTSTEP Graphical Interfaces to Control Sound Processing and Spatialization Instruments. *Proc. International Computer Music Conference*, Thessaloniki, Greece, 1997.

[24] D. Topper, M. Burtner, S. Serafin. Spatio-Operational Spectral (S.O.S.) Synthesis. *Proc. International Conference on Digital Audio Effects (DAFx-02)*, Hamburg, Germany, 2002.