LE " SCENE-MODELER " : DES OUTILS POUR LA MODELISATION DE CONTENUS MULTIMEDIAS INTERACTIFS SPATIALISES

Tifanie Bouchara Bouchara

LE « SCENE-MODELER » : DES OUTILS POUR LA MODELISATION DE CONTENUS MULTIMEDIAS INTERACTIFS SPATIALISES

Tifanie Bouchara

LIMSI/CNRS
Université de Paris XI, Orsay
tifanie.bouchara@limsi.fr

« Pour tirer parti des possibilités nouvelles, une recherche véritablement scientifique est donc nécessaire – recherche sonore et musicale, recherche pour la musique plutôt que sur la musique. »

J-C Risset

RÉSUMÉ

L'ensemble d'outils *SceneModeler* proposé ici concerne les environnements virtuels et la création artistique musicale. Il s'agit d'un *package* pour la création d'environnements interactifs temps-réel, immersifs, à la fois sonores et visuels.

Après la présentation des enjeux, nous présenterons la structure du *package* et ensuite la description de *Monthey04*, une proposition artistique, qui a permis la validation des applications développées.

Bien que fondé sur un exemple particulier, le *SceneModeler* peut être adapté à d'autres projets similaires, car il s'agit d'une suite logicielle ouverte, paramétrable et générique.

Mots-clés

Environnement immersif, interaction son et image, audio 3D, modélisation graphique 3D, partition navigable.

1. INTRODUCTION

Fortement influencés par le cinéma et la télévision, certains compositeurs cherchent aujourd'hui à allier leurs compositions musicales à des créations visuelles, soit comme illustration de la musique soit comme un nouveau matériau pour la composition ([5], [6]). L'heure est également à la recherche de nouveaux modes de jeu, de nouvelles interfaces musicales ([2]) et d'une meilleure intégration de l'espace sonore dans les compositions. C'est ainsi que les études sur les partitions navigables se développent (logiciel *EgoSound* [10], projet ENIGMES [4], interface *SoundEngine* [14]). Il s'agit, en effet, de nouveaux dispositifs d'interface musicale où le compositeur/utilisateur peut modifier en temps réel le rendu musical en se « promenant » dans un univers audiovisuel tridimensionnel. L'environnement graphique et l'interface utilisateur permettent alors le contrôle du rendu sonore ([1]).

Il est alors important de proposer des outils permettant de créer des installations interactives audio-visuelles. Le projet *InstallaSon* d'Antonio de Sousa Dias ([12], [13]) et le logiciel *EgoSound* en sont des exemples.

C'est dans cette optique de développer une application d'aide à la programmation de scènes multimédias interactives 3D que nous avons décidé de nous appuyer sur un projet multimédia : *Monthey04*. Ce projet se présente sous la forme d'un espace navigable dans un environnement virtuel 3D sonorisé. A chaque élément visuel de la scène correspond un élément sonore. L'utilisateur, que l'on peut appeler navigateur ou explorateur, est ainsi amené à découvrir par lui-même l'espace créé.

C'est donc pour mettre en place *Monthey04* que nous avons conçu le package *SceneModeler* pour la programmation de scènes multimédia spatialisées (son et graphique), composé de deux parties : un descripteur de scènes virtuelles et un outil de spatialisation sonore. Deux logiciels ont alors été utilisés : Virtual Choreographer (VirChor)[1] ([7], [8]) pour la partie visuelle, Max/MSP ([17], [18]) pour le son. Le *SceneModeler* est flexible et paramétrable tant en termes de programmation de la scène qu'en termes de communication.

En effet, en plus de la communication entre les deux logiciels, il fallait également prendre en compte l'utilisateur. En effet, l'intérêt de ce genre d'application est que ce dernier peut contrôler la scène et se déplacer dans l'environnement de façon simple.

Ainsi, *Monthey04* a été l'occasion de tester les outils développés.

2. UN POINT DE DÉPART : MONTHEY04

Pour établir le *SceneModeler*, nous sommes partis d'une proposition de contenu à mettre en oeuvre. Cette partie artistique a été entièrement prise en charge par Antonio de Sousa Dias et a été nommée *Monthey04*.

L'idée de base est la suivante : il s'agit de prendre l'expérience radiophonique « Points d'écoute, points de vue » proposée par le compositeur Pierre Mariétan ([11]) et de construire une installation multimédia. Cette expérience a été menée à Monthey (Suisse) du 28 août au 5 septembre 2004 lors des 7èmes rencontres A.M.E.04, en association avec ASM. Dans cette expérience, 5 participants ont été répartis en 5 lieux. Chaque

[1] Développé au LIMSI par Christian Jacquemin, VirChor est distribué librement sur http://virchor.sourceforge.net.

participant a passé 24h sur un lieu et fait un bulletin radiophonique rendant compte de ses impressions. Le lendemain les participants échangeaient leur place de sorte qu'au bout de 5 jours, les 5 participants aient vu les 5 lieux. 25 interviews témoignent donc de ce travail.

De façon symbolique, dans *Monthey04*, des extraits sonores du projet radiophonique sont associés à des objets visuels géométriques. Les interviews sont représentées par des carrés. Ceux-ci sont répartis en 5 étages symbolisant les lieux de la Suisse qui ont été visités, et ce par altitude. De plus, les carrés sont également répartis verticalement par journée. Au centre, on aperçoit 3 sphères : l'une porte le jingle radio qui séparait chaque interview de l'expérience radiophonique, les deux autres la voix du commentateur (Figure 1).

Le but de cette œuvre est que l'utilisateur puisse se déplacer à l'intérieur de la scène. Plus il se rapproche d'une image (un carré), plus il peut entendre l'extrait sonore qui lui est associé. Les éléments sonores sont des extraits des interviews initiales « Points d'écoute, points de vue ». C'est donc au gré de son voyage que le spectateur est amené à découvrir les différents commentaires.

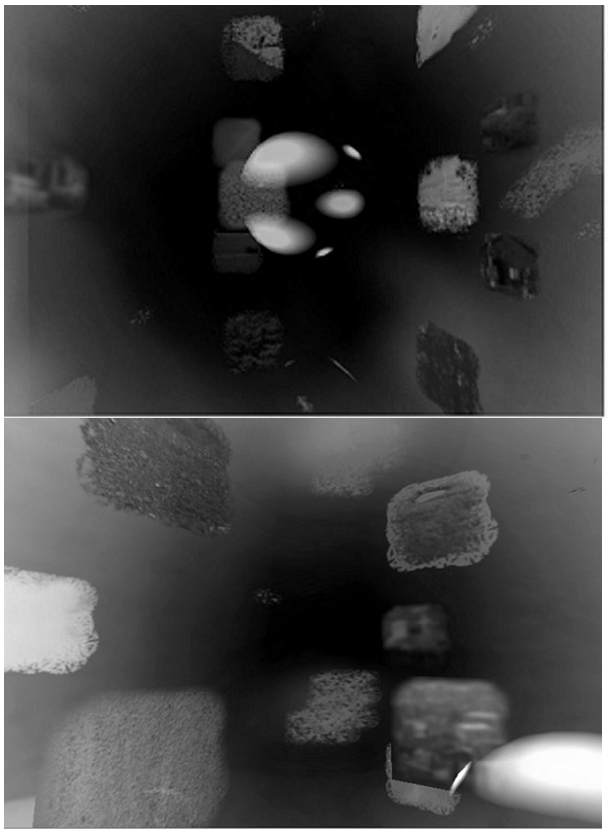

Figure 1. Deux vues différentes présentant le rendu visuel de la scène de *MontheyO4*.

L'intérêt de ce projet est qu'il permet d'associer une image à ces extraits sonores, et en particulier d'associer ce que les interviews n'évoquaient pas dans le projet radiophonique : la couleur. Ainsi chaque carré est texturé par une photo retouchée du lieu visité. Les bords de ces carrés sont volontairement flous et transparents de façon à gommer les formes géométriques initiales (Figure 2).

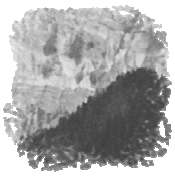

Figure 2. Une des textures employées pour les carrés.

3. DESCRIPTION DU SCENE-MODELER

3.1. Architecture globale du projet

Le *SceneModeler* permet de réaliser des scènes interactives en temps-réel, audiovisuelles et spatialisées. L'architecture générale de chaque projet développé est présentée en Figure 3. On remarque que cette architecture est triangulaire et s'organise autour de 3 pôles : un utilisateur, une composante graphique et une composante audio. Chaque « sommet » ayant une relation d'interaction avec les deux autres.

Deux applications complémentaires sont utilisées. Une application, implémentée sous l'environnement graphique Max/MSP, permet un rendu sonore spatialisé. Tandis que VirChor est exploité pour deux de ses fonctionnalités : le rendu graphique 3D d'une part, la partie commandes de l'autre.

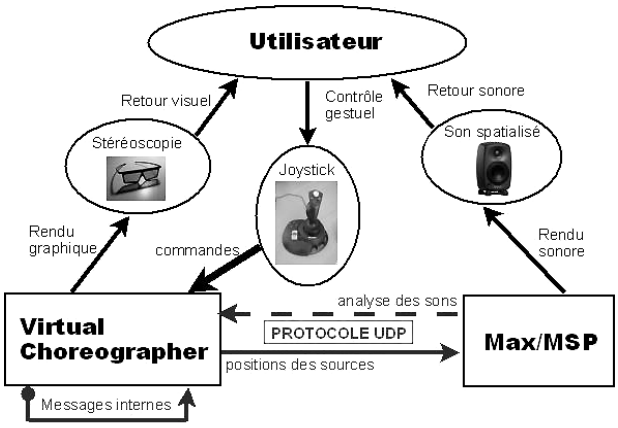

Figure 3. Architecture globale du projet.

3.2. Communication inter-logicielle

L'architecture présentée ci-dessus nécessite de développer un système communicant entre les différentes applications. Le dispositif actuel utilise une seule machine pour toutes les applications avec des communications internes. Mais on pourrait aussi utiliser une architecture distribuée sur plusieurs machines afin de répartir les tâches (un ordinateur pour le rendu audio, un autre pour le rendu graphique) et ainsi diminuer les temps de calcul.

L'ensemble des communications réseaux entre les logiciels se fait par des messages au format OSC

(OpenSoundControl[2] [15]) par protocole UDP (User Datagram Protocole). D'une certaine manière, l'OSC en tant que méthode de transport d'informations pour le contrôle est un remplaçant du MIDI.

L'avantage de cette méthode est que les deux logiciels VirChor et Max/MSP peuvent tous les deux être à la fois serveurs et clients. Serveurs dans le sens où ils peuvent recevoir des messages envoyés sur le réseau par l'autre application, clients car ils peuvent également envoyer des messages.

Pour VirChor ce module de communication UDP est intégré au logiciel. Mais, lorsque nous avons développé les outils du *SceneModeler*, Max/MSP ne contenait pas ces fonctionnalités et il était nécessaire d'utiliser une libraire externe. Nous avons utilisé « netsend » et « netreceive »[3]. Maintenant, ces fonctionnalités sont intégrées dans Max/MSP.

3.3. Modélisation graphique 3D interactive

Pour la modélisation de la scène nous avons utilisé VirChor, un navigateur de scènes 3D Open Source basé sur le langage XML et qui s'appuie sur la librairie de rendu graphique multi-plateforme OpenGL.

Le fonctionnement du logiciel est le suivant : on modélise une scène par des noeuds (des balises) qui forment un graphe de scène. La Figure 4 présente un exemple de ce type de graphe dont la partie en gras est codée en Figure 5.

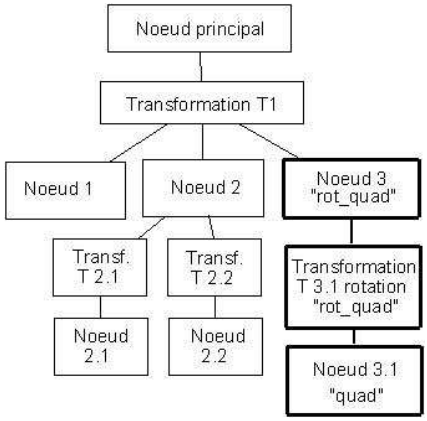

Figure 4. Exemple de graphe de scène.

```
<node id = "rot_quad" >
<transformation id="rot_quad"
     geometry="rotation"
     angle = "50"
     x = "0" y = "1" z = "0"/>
  <node id = "quad">
     <use xlink:href = "quad.xml:#quad"/>
  </node>
</node>
```

Figure 5. Un exemple de code XML sous VirChor : le noeud rot_quad permet d'instancier un objet carré et de lui appliquer une rotation.

[2] http://opensoundcontrol.org/
[3] http://www.akustische-kunst.org/maxmsp/

On y définit les propriétés géométriques et graphiques des objets comme la forme, la position/orientation, les couleurs. Chaque scène modélisée doit présenter au moins un utilisateur et une caméra (point de vue et point d'écoute par défaut de l'utilisateur). Dans ce sens, on peut dire que la modélisation graphique s'organise comme dans tout logiciel de modélisation type 3DSMax ou Blender.

Cependant on trouve deux différences majeures entre VirChor et ces logiciels. La première est qu'il permet également de donner aux objets des propriétés sonores (fichier de son émis, niveau, nombre de boucles, et bien sûr la position liée au nœud). La scène 3D ainsi décrite possède donc plusieurs informations relatives à l'audio : VirChor ne contient pas son propre moteur de rendu audio mais il envoie en temps réel la position relative des sources sonores par rapport au point d'écoute. Deuxièmement, les scènes décrites peuvent être utilisées et modifiées en temps réel et présentent la possibilité de gérer des interactions soit entre l'utilisateur et la scène, soit entre les objets eux-mêmes. Cet aspect dynamique est intégré à la scène par des scripts (Figure 6). Cela fonctionne par envoi de messages entre les différentes composantes de la scène ou entre applications (par exemple entre VirChor et Max/MSP). Avec ce système de communication, le rendu visuel et sonore est calculé en temps réel via les deux logiciels.

```
<node id = "root_joystick_capture">
 <script id = "script
 joystick_capture">
  <command>
   <trigger
   type = "cyclic_time"
   begin = "0"
   period = "0.00002"
   state="active"
   bool_operator="=="/>
    <action>
        <set_node_scalar_value
        id = "angleyaw"
        value = "({$configuration
        node:joystick_x}/600)"
        operator = "-="/>
          <target
          type = "single_node"
          value = "#root"/>
    </action>
  </command>
 </script>
</node>
```

Figure 6. Exemple de script et de commande : changement du paramètre "angleyaw".

Enfin, l'effet de profondeur est accentué par l'utilisation d'un système de projection stéréoscopique qui renforce d'autant plus l'immersion de l'utilisateur.

3.4. Traitement de l'audio 3D

Dans le *SceneModeler*, nous avons employé Max/MSP comme moteur de rendu audio spatialisé. Une application, *SceneEncDec*, permet de placer, selon les informations reçues de VirChor et de façon automatique, les sources sonores dans l'espace. En particulier, elle reçoit les données de position de chaque source (azimut, élévation et distance) par rapport au point de vue

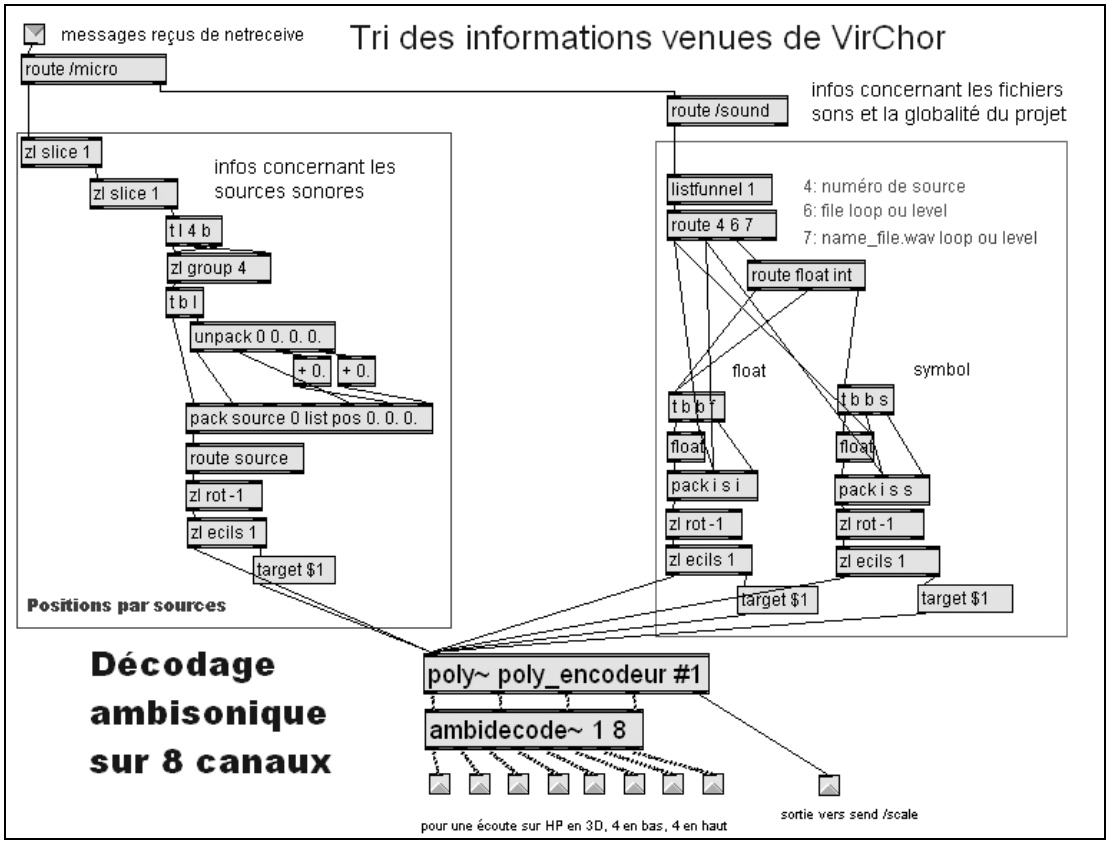

Figure 7. Le patch qui gère le traitement du son (encodage et décodage ambisonique) dans l'application *SceneEncDec* développée sous Max/MSP.

(camera) dans la modélisation géométrique. Ainsi, elle engendre un rendu audio cohérent avec les objets, qu'ils soient visibles (en face) ou non (hors-champ visuel), et ce sur un espace à 360°.

Pour la spatialisation, nous nous sommes appuyés sur le système Ambisonic, à l'ordre 1 (B-format). Cela permet d'avoir un système d'écoute flexible et modulable pour d'autres configurations concernant le nombre et la disposition des haut-parleurs. Le calcul du rendu 3D sonore se fait donc au travers des Ambisonics externals de G. Wakefield[4] et aux objets de l'ICST[5] ([9]). Ainsi chaque source est d'abord encodée au B-format selon sa position, puis décodée sur une disposition de haut-parleurs préalablement choisie (Figure 7).

Un grand nombre de sources peuvent être activées en même temps. Le rendu sonore est dépendant de l'ensemble de ces sources dont la spatialisation doit être gérée en temps-réel. L'implantation du spatialisateur ambisonique se fait donc grâce à l'objet *poly~*. En effet, il utilise plusieurs instances du même modèle d'encodeur et permet de désactiver dynamiquement les instances non utilisées. On gagne ainsi en ressources ce qui favorise la gestion temps-réel de la spatialisation. De plus, cela permet d'encoder de façon indépendante toutes les sources sonores. L'encodage est réalisé dans l'abstraction *poly_encodeur (*Figure 8). Le décodage, lui, se fait de la même manière pour toutes les sources car il dépend uniquement de la disposition des enceintes. Pour le projet *Monthey04*, nous avons opté pour un système de 8 haut-parleurs répartis en cube autour de l'auditeur avec l'écran sur une face.

De plus, le *SceneModeler* contient également un objet *calibration* qui calibre automatiquement le niveau des 8 enceintes utilisées, assurant que le système de diffusion ne nuit pas au traitement ambisonique effectué.

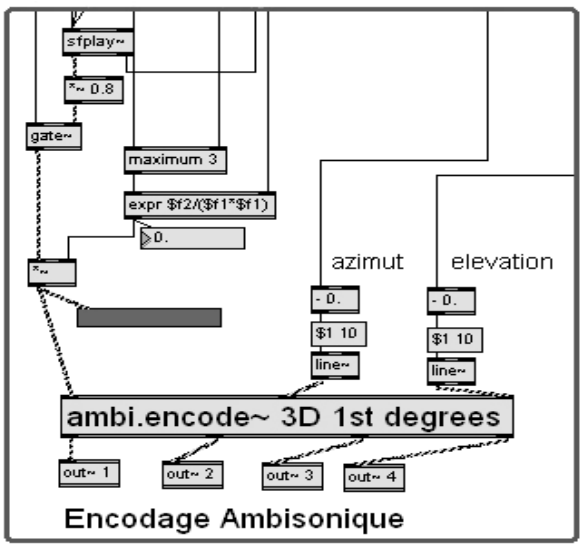

Figure 8. Un extrait du *poly_encodeur* : encodage ambisonique pour chaque source sonore.

[4] www.grahamwakefield.net/soft/ambi~/index.htm
[5] www.icst.net

Finalement, dans la volonté d'obtenir un ensemble qui soit portable et utilisable avec le moins de matériel possible, nous avons décidé d'intégrer la possibilité pour l'utilisateur d'utiliser un casque et de profiter d'un rendu spatialisé par un décodage de l'ambisonique vers un rendu binaural, d'où la création d'une seconde application *SceneEncDec_BINO* où la partie décodage ambisonique est remplacée par un décodeur binaural *ambi2bin~* créé au LIMSI.

3.5. Communication et interactions homme-machine

La communication entre l'utilisateur et l'application se fait au travers d'un joystick. Ce dernier permet à l'utilisateur de naviguer dans l'espace virtuel. Ainsi l'utilisateur devient explorateur.

La possibilité pour VirChor de recevoir des informations liées à un joystick a été intégrée au logiciel lors de la création du projet *Monthey04*. Ainsi il suffit lors de la création de la scène virtuelle d'ajouter à la scène le fichier *Joystick_Capture.xml* qui reçoit les informations utiles concernant le mouvement du joystick.

Pour qu'il ne soit pas intrusif, ce dernier devait avoir un fonctionnement intuitif. Un usage proche de celui qu'on en a dans les jeux de simulation de vol a été choisi puis adapté à la scène créée. Plusieurs méthodes de navigation ont été explorées. Celle qui a été retenue pour *Monthey04* consiste en:

- des mouvements avant/arrière du joystick, ils permettent de pencher légèrement la caméra vers le bas/haut (tilt) mais également de se déplacer selon l'axe vertical (travelling vertical)
- des mouvements latéraux gauche/droite, ils permettent une rotation panoramique de la caméra.
- des boutons qui permettent d'avancer ou reculer la caméra (travelling avant/arrière) à vitesse variable.

Dans une autre expérimentation, la caméra ne devait plus pouvoir traverser les objets. Un système de détection de contact a alors été mis en place et le mouvement de la caméra a été adapté pour qu'elle « glisse » sur les objets rencontrés. Cette fonctionnalité n'a pas été employée dans *Monthey04* mais pourra être utilisée dans d'autres projets.

L'utilisateur reçoit, lui, un retour visuel et sonore lié à ces mouvements (comme le son donné par un instrument renseigne le musicien sur son jeu). On peut dire que l'interaction se fait aussi du rendu graphique et sonore vers l'utilisateur. L'interaction temps réel (et une grande corrélation spatiale et temporelle des évènements) permet une meilleure immersion de l'utilisateur et, de ce fait, favorise son engagement dans le dispositif proposé ([3]).

3.6. Interaction multimodale

La navigation à travers l'espace virtuel visuel permet de créer l'espace sonore : les objets sonnent lorsque la caméra, vue du spectateur, s'approche des objets. Cette bimodalité audio-visuelle est proche de celle que l'on rencontre dans notre vie quotidienne : elle donne aux objets sonores spatialisés des comportements tantôt physiques tantôt symboliques. La représentation visuelle fait alors fonction de repérage. Elle informe sur les potentialités sonores des objets observés. Tandis que la position perçue auditivement permet de reconstruire l'espace lorsque les objets ne sont pas ou plus visibles. Nous sommes donc en présence d'une double interaction puisque l'image agit sur l'intention d'écoute et que le son agit sur la perception de la perspective ressentie de la scène.

De plus, comme H. Zenouda ([16]) et d'autres l'avaient déjà remarqué, l'interactivité modifie les rapports entre l'image et le son. Notre interprétation du son, et de sa spatialisation, ne dépend plus comme au cinéma des mouvements de l'image (son in/off/hors champ) car la cohérence dans les déplacements des objets sonores hors-champ visuel est augmentée par l'interactivité en temps-réel. Ainsi l'interaction permet d'insérer une valeur ajoutée dans la compréhension que l'on tire de l'image et du son. Le son et l'image fusionnent dans un nouveau procédé où il faut penser le processus en amont.

4. PERSPECTIVES

Le *SceneModeler* fonctionne pour le moment avec les objectifs que l'on s'était fixés au départ. Cependant, de façon à augmenter la cohérence entre l'image et le son ainsi que la perception des liens entre les objets et les extraits sonores entendus, nous avons décidé d'apporter plusieurs fonctionnalités supplémentaires.

La première sera de permettre une influence du son sur l'image. Ainsi l'analyse des sons sera utilisée pour faire varier les paramètres géométriques des objets. Par exemple, lorsque un objet est suffisamment proche de l'utilisateur, le couple image/son est sélectionné. L'image montre alors cette sélection en rendant plus vives les couleurs de l'objet. Ainsi le spectateur perçoit immédiatement que le son écouté est associé à l'image vive. D'autres paramètres (intensité, attaque et hauteur pour le son ; taille, position et transparence pour l'image) peuvent également être exploités et donc augmenter les interactions entre les médias.

De plus, dans la volonté de mettre en place une installation de diffusion, il serait intéressant de développer un dispositif de captation de geste qui permettrait non plus à un utilisateur d'explorer l'espace visuel et sonore proposé mais bel et bien à un danseur ou compositeur de faire voyager le spectateur dans ce monde. De la même manière, il a également été évoqué la possibilité de mettre en place un système plus important avec une captation de position et où le déplacement à l'intérieur du monde virtuel se ferait en fonction des positions et des déplacements de l'ensemble des spectateurs, ou plus exactement des visiteurs présents à l'intérieur de l'espace physique créé par le dispositif.

Le rapport réaliste pourrait être lui aussi augmenté en modélisant par la suite la directivité des sources sonores et des microphones virtuels dans l'environnement. Il est aussi envisagé de créer des objets ayant des propriétés acoustiques remarquables telles que absorption,

réflexion, dispersion. Ceci permettrait notamment la création de « murs acoustiques », présentant des intérêts artistiques autant que scientifiques.

5. CONCLUSIONS

Le projet *Monthey04* a permis de voir comment les éléments visuels et sonores pouvaient s'agencer. Grâce à au *SceneModeler*, regroupant des outils graphiques et sonores, Antonio de Sousa Dias ainsi que d'autres compositeurs pourront « illustrer » leurs compositions musicales et laisser s'exprimer leur talent et leur imagination au travers d'outils et de techniques maintenant plus faciles à mettre en place. Génériques, les outils du *SceneModeler* devraient faire de *Monthey04* le premier projet d'une longue série.

6. REMERCIEMENTS

Je tiens à remercier Christian Jacquemin, Brian FG Katz, Antonio de Sousa Dias et Anne Sedes pour m'avoir permis de travailler sur ce projet extrêmement intéressant et enrichissant et pour m'avoir aussi bien encadrée.

7. REFERENCES

[1] Afonso, A., Blum, A., de Laubier, S., Denis, M., Folch, H., Genevois, H., Katz, B., Nugier, S., Schnell, N. « Design d'environnements multimodaux interactifs communicants ». *In proceedings, Hypermedias Hypertexts, Products, Tools and Methods H2PTM'05*, Paris, France, 2005.

[2] Blue Yeti , « Le dessin musical: voyage entre art, science et technologie », *Captures 16.* http://www.blueyeti.fr/IMG/pdf/captures16.pdf

[3] Bouvier, P., Loyet, R., Chaudeyrac, P., Piranda, B., de Sorbier de Pougnadoresse, F. « Immersion dans un monde visuel et sonore en 3D », *Journées de l'Association Francophone d'Informatique Graphique*, 2006.

[4] Cahen, R. « Compte rendu du projet ENIGMES (Expérimentation de Nouvelles Interfaces Graphiques d'expression Musicales et Sonores) », 2007. projetenigmes.free.fr

[5] Courribet, B. "Réflexions sur les relations musique/vidéo et stratégies de mapping pour Max/MSP/Jitter", *Actes des douzièmes Journées d'Informatique Musicale*, AFIM/CICM-Paris8/MSH Paris Nord, Paris, 2005.

[6] Dannenberg, R. B. "Interactive Visual Music : A Personal Perspective", Computer Music Journal : Vol. 29, No. 4, winter 2005. Cambridge, MA : MIT Press. pp 25-35.

[7] Jacquemin, C. "Architecture and Experiments in Networked 3D Audio/Graphic Rendering with Virtual Choreographer", *Proceedings, Sound and Music Computing (SMC'04)*, Paris, 2004.

[8] Jacquemin, C. *Virtual Choreographer Reference Guide (version 1.4)*, LIMSI-CNRS et Université Paris11. http://virchor.sourceforge.net/html/index.html (15/01/08)

[9] Schacher, J. C., Kocher, P. "Ambisonics Spatialization Tools for Max/MSP", *Proceedings of the 2006 International Computer Music Conference*, New Orleans, 2006

[10] Sedes, A., Courribet, B., Thiébaut, J-B. « visualisation du sonore avec le logiciel egosound : WORK IN PROGRESS », *Actes des dizièmes Journées d'Informatique Musicale*, Metz, 2003.

[11] Sousa Dias, A. (org.) « Dossier : Points d'écoute, points de vue », *Sonorités – Chronique de la chose entendue*, n.2, Champ Social Éditions, (prév. 2008).

[12] Sousa Dias, A. « InstallaSon–KITTy :vers le développement d'outils d'assistance à la conception et à la construction d'espaces musicaux navigables » – *Rapport 2007*, FCT/MCTES, Portugal, nov. 2007.

[13] Sousa Dias, A. « InstallaSon : un éditeur et gestionnaire d'espaces musicaux navigables ». *Actes des Journés d'Informatique Musicale 2008*, Albi, 2008.

[14] Wozniewski, M., Settel, Z., Cooperstock, J. R. "A spatial interface for audio and music production", *9th Conference on Digital Audio Effects*, September 2006.

[15] Wright, M., Freed, A., Momeni, A. "OpenSound Control : State of the Art 2003", *Proceedings of the 2003 Conference on NIME*, Montreal, Canada, 2003.

[16] Zenouda, H. « Relation image/son : de l'illustration sonore à la fusion multi-modale », *2ième biennale internationale "Autour de l'illustration, penser les images"*, Bobigny, novembre 2006.

[17] Zicarelli, D. "An Extensible Real-Time Signal Processing Environment for Max", *Proceedings of the International Computer Music Conference 1998*. International Computer Music Association, Ann Arbor, 1998, pp 463-466.

[18] Zicarelli, D., Taylor, G., Clayton, J. K., Dudas, R., Nevile B., MAX 4.6 : Reference Manual http://www.cycling74.com/