

SPATIALISATION SONORE APPLIQUEE A  
UN SYSTEME DE REALITE VIRTUELLE  
CNRS LMA Marseille - CRVM Luminy

Patrick Marmaroli

janvier 2008

## **Remerciements**

Je remercie toute l'équipe du LMA pour son aide et sa contribution à une excellente ambiance de travail, en particulier M. Richard Kronland-Martinet et M. Thierry Voinier pour leur encadrement, leur disponibilité et leur pédagogie.

Je remercie l'équipe du CRVM de Luminy pour leur accueil, en particulier M. Daniel Mestre, M. Stéphane Gobron, M. Jean-Marie Pergandi et M. Cédric Goulon pour leurs conseils et leur collaboration.

## Résumé

Le Centre de Réalité Virtuelle de la Méditerranée (CRVM, Marseille) est équipé d'un système de visualisation stéréoscopique destiné à immerger sensoriellement un utilisateur. Cette plate-forme, ou CAVE (Cave Automatic Virtual Environment) est composée de 3 écrans de 3X4 mètres (devant et sur les côtés) et d'un plateau de 3X3 mètres (au sol). L'opérateur, placé au centre du dispositif est plongé dans une scène virtuelle et est équipé de capteurs permettant de déterminer en temps réel la position de ses segments corporels (tête, bras, main, jambes, tronc, regard, etc). Ces informations permettent à l'opérateur d'interagir avec l'environnement virtuel qui l'entoure.

Le but de ce stage est de coupler à cet environnement visuel un environnement sonore 3D. Ceci via le développement d'un logiciel qui réceptionne les coordonnées de l'utilisateur (position, orientation), les convertit et les communique à un autre logiciel dit de "spatialisation sonore" de manière à ce que le son, tout comme l'image, évolue en fonction du comportement de l'utilisateur.

La spatialisation sonore est une technique permettant de traiter les sons de manière à les rendre perceptibles dans l'espace, on parle alors de "sons 3D" ou "sons spatialisés". Ainsi on peut donner à l'auditeur l'impression d'une source sonore provenant d'une distance, d'une élévation et d'un azimut fixés, bien que le son provienne en réalité d'un casque audio. L'intérêt principal est d'accroître l'immersion de l'utilisateur au sein du CAVE.

## **Abstract**

The Mediterranean Virtual Reality Center (MVRC, Marseille) is equipped with a stereoscopic visualisation system aiming at producing an immersive sensory simulation. This platform is called CAVE : Cave Automatic Virtual Environment and consists in three  $3 \times 4$  meters screens (one in front and one on each side) and a  $3 \times 3$  meters screen on the floor. The user - located in the center of this system and immersed in a virtual scene - is equipped with sensors which can determine, in real times, the position of his bodily segments (head, arms, hands, legs, torso, look etc.). This information enables interactions between the user and the virtual environment around him.

The goal of this training course is to link a 3D sonic environment with the visual environment. This was done through the implementation of a software that receives the user's coordinates (position and orientation) and converts them in order to transmit them to a "sound spatialisation" software. This enables the sound to evolve according to the images and to the user's behaviour.

The sound spatialisation is a technique which allows to reproduce sounds while giving the possibility to localize them in space. It is also called "3D sounds" or "spatialized sounds", and the produced sound effect gives the user the impression of sounds which come from a certain distance, azimuth and elevation, even if they come from a headphone. The main purpose of this technique is to increase the immersivity of the user in the CAVE.

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>7</b>
<b>2</b>	<b>Présentation du stage</b>	<b>8</b>
2.1	Les lieux du stage . . . . .	8
2.1.1	Le Laboratoire de Mécanique et d'Acoustique . . . . .	8
2.1.2	Le Centre de Réalité Virtuelle de la Méditerranée . . . . .	11
2.2	Descriptif du stage . . . . .	13
<b>3</b>	<b>Contraintes et outils utilisés</b>	<b>15</b>
3.1	MAX/MSP . . . . .	15
3.2	Spat . . . . .	15
3.3	Contraintes techniques . . . . .	16
<b>4</b>	<b>Eléments de perception acoustique</b>	<b>17</b>
4.1	Les indices acoustiques de localisation . . . . .	18
4.2	Les HRTF . . . . .	21
4.3	Simulation du mouvement : effet Doppler . . . . .	23
4.4	Impression de distance . . . . .	25
4.5	Conclusion sur les éléments de perception acoustique . . . . .	26
<b>5</b>	<b>Présentation de ViSpat</b>	<b>27</b>
5.1	Protocole de communication . . . . .	28
5.2	La gestion des coordonnées . . . . .	31
5.3	Gestion de l'orientation des sources dans le plan horizontal . . . . .	34
5.4	Gestion de l'azimut et de l'élévation . . . . .	34
5.5	L'interface utilisateur . . . . .	35
5.6	La gestion d'erreurs . . . . .	36
5.7	La lecture des fichiers sons . . . . .	37
5.8	Tests avant installation . . . . .	37
<b>6</b>	<b>Conclusion et perspectives</b>	<b>38</b>
6.1	Une limite de Spat . . . . .	38
6.2	La démonstration . . . . .	39
6.3	Conclusion et perspectives . . . . .	40

<b>7</b>	<b>Annexes</b>	<b>41</b>
7.1	Fiche Technique ViSpat . . . . .	41
7.1.1	Traitement des trames . . . . .	41
7.1.2	Traitement de la trame auditeur . . . . .	45
7.1.3	Gestion de l'orientation des sources dans le plan horizontal . . . . .	47
7.1.4	Gestion de l'azimut et de l'élévation d'une source . . . . .	48
7.2	patch Main . . . . .	52
7.3	patch RecepteurUDP . . . . .	53
7.4	patch RecupRTa . . . . .	53
7.5	patch AiguillageRTs . . . . .	54
7.6	patch RecupRTs . . . . .	54
7.7	patch YawPitch . . . . .	55
7.8	patch TraitementEspace . . . . .	56
7.9	patch TraitementFormat . . . . .	57
7.10	patch TraitementTrames . . . . .	58
7.11	patch Ambiance2Buff . . . . .	59
7.12	patch So2Buff-Syn1 . . . . .	60
7.13	patch Buffer . . . . .	61
7.14	patch ConvCooPos . . . . .	62
7.15	code Virotation . . . . .	62
7.16	patch SoSpat . . . . .	65
7.17	patch SpatControl . . . . .	66
7.18	patch CooControl . . . . .	66
7.19	patch SpatReverb . . . . .	67
7.20	patch LecS . . . . .	68
7.21	patch AcoustiqueGénérale . . . . .	69
7.22	patch Error000 . . . . .	70
7.23	patch Error100 . . . . .	70
7.24	patch Error200 . . . . .	71
7.25	patch Error400 . . . . .	71
7.26	patch Error500 . . . . .	72
7.27	patch Error600 . . . . .	73
7.28	patch Error700 . . . . .	73
7.29	patch Error800 . . . . .	74
7.30	patch Error900 . . . . .	74
7.31	Fiche d'utilisation ViSpat . . . . .	75
<b>8</b>	<b>Bibliographie</b>	<b>76</b>
<b>9</b>	<b>Références Sites</b>	<b>77</b>

# Table des figures

2.1	Le système immersif en position ouverte (à gauche) et semi-fermée (à droite). Les écrans de projections verticaux sont de 3X4 mètres. . . . .	12
4.1	Système de représentation sphérique de l'espace par rapport à la tête où r est la distance, d est l'élévation et j est l'azimut [d'après Blauert] . . . . .	18
4.2	Utilisation de l'ITD pour la localisation en azimut. [d'après Cheng and Wakefield (2001)] . . . . .	19
4.3	Mesure et simulation d'HRTF . . . . .	21
5.1	Fonctionnement général de ViSpat . . . . .	27
7.1	Schéma du traitement de la trame d'initialisation . . . . .	42
7.2	schéma du traitement de la trame source . . . . .	44
7.3	schéma du traitement de la trame auditeur . . . . .	45
7.4	Principe d'asservissement de ViSpat . . . . .	47
7.5	Schéma d'une rotation sur X puis sur Z puis sur Y . . . . .	49

# Chapitre 1

## Introduction

Le Stage d'Application Ingénieur (SAI) est une étape essentielle dans le cursus d'un étudiant. Il permet de mettre en pratique nos connaissances dans un milieu professionnel et d'acquérir une première expérience du métier d'ingénieur. Pendant une durée minimale de trois mois, l'étudiant doit élaborer un projet demandé par l'entreprise et présenter ses travaux devant un jury à la fin du stage.

Ayant opté pour la spécialisation de cinquième année "Traitement de signal et Trajectographie", je me suis orienté vers un stage de quatrième année liant à la fois ma passion pour la musique et l'envie d'évoluer dans un milieu de recherche. J'ai donc travaillé avec l'équipe du Laboratoire de Mécanique et d'Acoustique (LMA) du CNRS de Marseille sur un projet de son 3D appliqué à un système de réalité virtuelle.



## Chapitre 2

# Présentation du stage

### 2.1 Les lieux du stage

#### 2.1.1 Le Laboratoire de Mécanique et d'Acoustique

##### Présentation générale

Cette présentation est inspirée des textes officiels du site internet du Laboratoire de Mécanique et d'Acoustique (LMA) (voir 1 page 77).

Le Laboratoire de Mécanique et d'Acoustique (LMA) a été fondé en 1941 à partir des moyens du Centre d'Etudes de la Marine Nationale. Initialement "Centre de Recherches Scientifiques Industrielles et Maritimes", il est devenu par la suite "Centre de Recherches Physiques" en 1963, puis a adopté en 1973 son intitulé actuel "Laboratoire de Mécanique et d'Acoustique" pour s'adapter ainsi à l'évolution de ses axes de recherches.

Le LMA est une Unité Propre de Recherche (UPR), du département Sciences et Technologies de l'Information et de l'Ingénierie du CNRS. Il est associé aux universités de la Méditerranée (Aix-Marseille 2) et de Provence (Aix-Marseille 1) et a des liens étroits avec l'Ecole Centrale de Marseille (anciennement EGIM). L'effectif du laboratoire est d'environ 120 personnes (chercheurs, enseignants-chercheurs, ITA et doctorants).

Les installations spécialisées du LMA, notamment en acoustique, en font l'un des laboratoires universitaires français les mieux équipés dans son domaine.

## **Les activités de recherche**

Les activités de recherche au LMA sont réparties en 4 axes :

- Matériaux
- Ondes
- Structures
- Sons

C'est au coeur de ce dernier pôle que j'ai réalisé mon stage, plus précisément dans l'équipe "Modélisation, Synthèse et Contrôle des Signaux Sonores et Musicaux". Les actions menées dans le cadre de cette opération de recherche visent à la construction d'outils numériques pour la synthèse et le traitement des sons et de la musique. Elles poursuivent les travaux sur l'analyse-synthèse menés au laboratoire depuis une quinzaine d'années, tout en les étendant aux processus de diffusion et de spatialisation, ainsi qu'à la prise en compte des relations entre la nature des sons, la perception et la cognition. L'essentiel de ces travaux donne lieu à des réalisations fonctionnant en temps réel sous environnement MAX/MSP.

## **Collaborations et partenariats**

Le LMA développe des collaborations étroites avec les Universités et les Ecoles d'Ingénieurs de la région d'Aix-Marseille. Il accueille des enseignants-chercheurs qui y effectuent leur recherche, des doctorants qui y préparent une thèse et les membres du LMA participent activement à l'enseignement de la Mécanique et de l'Acoustique en France.

Partenaire de plusieurs contrats européens, développant de nombreuses coopérations à travers le monde, le laboratoire a également vu confier à plusieurs de ses membres des responsabilités importantes dans l'organisation de groupements de recherche ou de programmes internationaux de coopération scientifique. Il accueille chaque année plusieurs visiteurs étrangers seniors ou de niveau doctoral ou postdoctoral pour des séjours de longue durée, et de nombreux scientifiques pour des séjours de courte durée.

Le LMA participe activement à divers types de programmes (Groupements de Recherche français et européens, Actions Concertées Incitatives, Plan-Pluri Formation, Equipe de Recherche Technologique, contrats européens, . . .). Il développe des collaborations régulières avec les grands groupes industriels (dans le domaine des transports, de la sécurité nucléaire, . . .), ainsi qu'avec les PME/PMI locales.

Enfin, le laboratoire s'investit fortement dans la formation par la recherche, notamment au sein de l'Ecole Doctorale "Physique, Modélisation et Sciences pour l'Ingénieur" et de l'ECM.

## **L'équipe**

L'équipe "Modélisation, Synthèse et Contrôle des Signaux Sonores et Musicaux" du LMA est composé de

- 1 responsable : Richard Kronland-Martinet (Directeur de Recherche).
- 5 participants : Jean Kergomard (Directeur de Recherche), Philippe Guillemain, Solvi Ystad, Christophe Vergez (Chargés de Recherche), Thierry Voinier (Ingénieur de Recherche).
- 4 doctorants : Mathieu Barthet, Marie-Céline Bezat, Adrien Merer et Thibaud Necciari.

## **Perspectives**

L'équipe s'appuie aujourd'hui sur 2 opérations de recherche en forte interaction "Modélisation, Synthèse et Contrôle des Signaux Sonores et Musicaux" et "Physique des Instruments de Musique". La collaboration entre ces deux orientations s'articule autour de 3 axes fondamentaux de recherche :

- La création des sons (analyse, instruments numériques et mécaniques, spatialisation, codage...)
- Le contrôle des sons (synthèse, temps-réel, geste, interprétation...) : projet ANR CONSONNES.
- La perception et la cognition sonore (réalité virtuelle sonore, timbre, sémiotique des sons...) : projet ANR senSons.

Les travaux sur la synthèse temps-réel d'instruments de musique seront poursuivis. L'installation au laboratoire d'une bouche artificielle devrait permettre, après plusieurs années de développement d'algorithmes de synthèse, c'est à dire le problème direct, de s'intéresser de façon systématique au problème inverse, consistant à déterminer les paramètres du modèle par l'analyse de sons produits dans des conditions connues et reproductibles. Ce travail s'inscrira naturellement dans le cadre du projet ANR CONSONNES : CONTRôle de SONs instrumentaux Naturels Et Synthétiques. La synthèse des sons en vue d'applications à la réalité virtuelle sera également au centre des préoccupations de l'équipe. Il s'agira notamment d'étendre les possibilités de la plate-forme d'analyse-synthèse à la prise en compte de l'interaction entre excitateur et structure, et notamment du geste (gratter, frotter, souffler,...). L'ouverture vers les sciences cognitives et la prise en compte de la perception

dans la définition des processus de génération sonore seront également poursuivies. L'étude du masquage temps-fréquence sera poursuivie dans le cadre de la thèse de T. Necciari en collaboration avec l'OR "Acoustique Perceptive". La sémiotique sonore, qui est au cœur des préoccupations des chercheurs en acoustique audio et neurosciences devrait être un point d'attraction fort susceptible de permettre la structuration d'une véritable équipe pluridisciplinaire, les projets scientifiques y afférents étant pour l'essentiel décrit dans les objectifs du projet *senSons* : vers le sens des sons.

## **2.1.2 Le Centre de Réalité Virtuelle de la Méditerranée**

### **Présentation générale**

Cette présentation est inspirée des textes officiels du site internet du Centre de Réalité Virtuelle de la Méditerranée (CRVM) (voir 2 page 77).

L'étude des liens entre la perception et la motricité passe par la recherche de moyens de stimulation sensorielle et de mesure comportementale qui soient à la fois artificiels pour être mieux contrôlables, et le plus proche possible de la réalité afin de placer le sujet en conditions écologiques d'interaction avec un environnement virtuel.

Ce constat a fédéré les équipes de l'I.F.R. Marey, autour du développement d'une Plate-forme Technologique de Recherche, mobilisant les technologies de pointe de la réalité virtuelle. Dans ce cadre, le CRVM constitue, en France, la première plate-forme technologique de réalité virtuelle, dédiée à l'étude du comportement humain en situation immersive.

Il constitue un terrain de convergence et d'enrichissement réciproque entre les domaines des Sciences de la Vie, des Sciences et Technologies de l'Information et de la Communication et de l'Ingénierie.

Le CRVM se consacre à la recherche fondamentale et appliquée, la formation et la valorisation. Le projet a été financé par l'Université de la Méditerranée, le CNRS, le Conseil Régional PACA, le Conseil Général des Bouches-du-Rhône et la Ville de Marseille.

### **Principales caractéristiques de la plate-forme de réalité virtuelle**

La plate-forme est un système de visualisation immersive, stéréoscopique, possédant une haute résolution spatiale et temporelle, comprenant une partie centrale (projection frontale et sol) et deux parties latérales (projections latérales). Voir figure 2.1.

La projection des images est assurée par des projecteurs SIM5+ (Barco), d'une résolution de 1400 x 1500 pixels. Les écrans latéraux sont alimentés par renvoi sur miroir. La projection sur l'écran placé au sol est assurée par projection directe.

La vision stéréoscopique est assurée par les couples de projecteurs (un projecteur pour chaque œil) associés à une technologie INIFTEC de séparation colorimétrique. Le spectateur porte des lunettes équipées de filtres spéciaux.

Des développements informatiques permettent de générer des applications en Réalité Virtuelle, capables de gérer une multi-visualisation et de s'interfacer avec les mesures comportementales utilisées dans le contexte. Pour ce faire, l'équipe utilise le logiciel Virtools qui intègre un langage de script graphique, utile au prototypage rapide d'applications.

Un système de mesure du comportement et des mouvements du sujet, permettant d'assurer l'interactivité du système, dans le cadre d'expérimentations dans le domaine des Sciences du Mouvement. Il s'agit d'un système de tracking ART, basé sur un ensemble de quatre caméras situées en position haute dans le système de visualisation. Ce dispositif permet de suivre la position de la tête de l'observateur dans l'environnement et de mettre à jour les scènes visuelles en fonction de celle-ci, en temps réel (à 60 Hz), grâce à la configuration de capteurs passifs placés sur les lunettes qu'il porte, c'est ce même dispositif qui sera utilisé pour la spatialisation sonore.

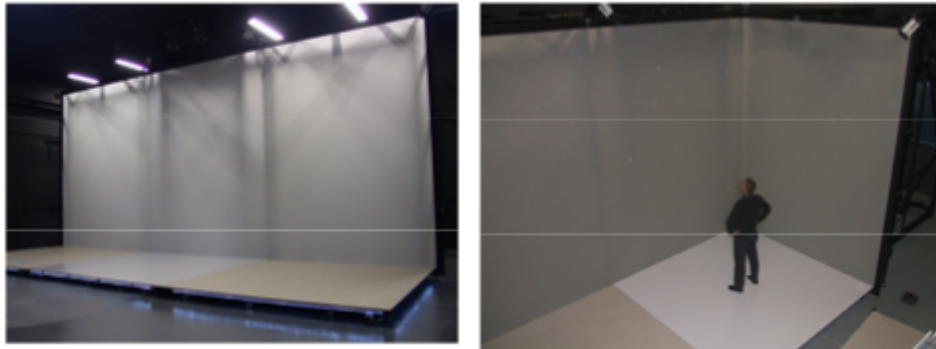


FIG. 2.1 – Le système immersif en position ouverte (à gauche) et semi-fermée (à droite). Les écrans de projections verticaux sont de 3X4 mètres.

## **L'équipe**

L'équipe Recherche et Développement en informatique 3D du centre du CRVM est composée de

- 1 responsable développement info-3D : Stéphane Gobron (ingénieur de recherche).
- 2 ingénieurs d'études : Jean-Marie Pergandi et Cédric Goulon.

L'ensemble des équipes du CRVM sont sous la responsabilité de Daniel Mestre, directeur de recherche CNRS.

## **2.2 Descriptif du stage**

### **Enoncé**

Le CRVM de Luminy dispose d'une plate-forme de réalité virtuelle : l'auditeur est "encerclé" d'écrans géants sur lesquels des images sont projetées, il porte des lunettes polarisées qui lui permettent de voir en 3D, ces lunettes sont équipées de capteurs qui renseignent le système informatique sur la position de l'auditeur et l'orientation de sa tête pour réajuster en temps réel les images, ainsi il a l'impression d'évoluer parmi les objets qu'il voit. Ce stage consiste à ajouter à ce système d'imagerie virtuelle un système de réalité virtuelle sonore. L'objectif étant d'accroître l'immersion de l'utilisateur par association du son à l'image. Le problème consiste à faire communiquer deux systèmes déjà existants mais dont les utilisations sont très différentes : le logiciel infographique de la plate-forme de réalité virtuelle du CRVM (développée pour le milieu industriel) et le logiciel de spatialisation sonore utilisé par le CNRS (développée pour les musiciens). Mon travail fut donc de développer un logiciel intermédiaire pour qu'ils puissent communiquer.

### **Déroulement du stage**

Mon travail s'est déroulé en 4 étapes.

Tout d'abord, j'ai dû acquérir une culture relative à la spatialisation sonore d'un point de vue psychoacoustique. Ce travail consiste en grande partie à lire et comprendre les rapports techniques, thèses et articles sur le sujet. Dans un second temps, j'ai suivi des cours de programmation sous environnement MAX/MSP, il s'agit d'un logiciel de programmation informatique destiné au traitement de signal audio (voir 3.1 page 15), par la suite il s'est avéré nécessaire d'étudier également le logiciel Spat qui est le spatialisateur sonore qui sera utilisé et qui fonctionne sous MAX/MSP (voir 3.2 page 15)

puis le logiciel Virtools qui est le logiciel utilisé par le CRVM pour commander les images 3D, gérer le scénario de chaque monde virtuel (positions des objets, réajustement des images en temps réel en fonction des mouvements de l'auditeur). J'ai ensuite développé l'application demandée sous environnement MAX/MSP. Enfin, j'ai quitté le CNRS de Marseille pour m'installer au CRVM de Luminy afin d'installer le matériel audio et le logiciel développé, d'en corriger les derniers bugs et de mettre en place une démonstration présentant le travail effectué.

## Chapitre 3

# Contraintes et outils utilisés

Cette partie présente les quelques outils informatiques sur lequel repose le projet ainsi que les différentes contraintes techniques qui justifient nos choix dans la manière de le réaliser.

### 3.1 MAX/MSP

MAX/MSP est le fruit de l'association du logiciel Max (Ircam/Cycling'74) et du logiciel MSP. Max a été conçu pour contrôler en temps réel les instruments MIDI, et MSP est une bibliothèque d'objets permettant le traitement de signal audio. Aujourd'hui MAX/MSP est utilisé pour des installations interactives, des performances sonores, une approche mathématique de la composition et recherche sonore ou bien pour des concerts en live. C'est sous l'environnement MAX/MSP que le logiciel demandé à été développé. *Pour plus d'informations sur MAX/MSP, consulter le site 5 page 77.*

### 3.2 Spat

Spat est un logiciel de spatialisation sonore basé sur une librairie de traitement de signal écrite en C++. Chaque module est disponible comme objet basique qui s'interconnecte avec MAX/MSP. Avec ce logiciel, l'utilisateur peut spécifier/automatiser les paramètres de spatialisation d'une source sonore indépendamment du mode de restitution ou du dispositif électroacoustique choisis. La spatialisation peut être accomplie en temps réel en utilisant un système de reproduction sonore basique (ex. reproduction binaurale avec un casque). Spat permet également de contrôler l'effet de salle (réverbération) en employant des critères perceptifs. Ce projet est né en 1991 d'une collaboration entre Espaces Nouveaux et l'Ircam, il inclut également les recherches menées à l'ENST sur les algorithmes de traitement de signal numérique pour la spatialisation et la réverbération artificielle de sons. *Pour plus d'informations sur Spat, consulter le site 6 page 77.*



### 3.3 Contraintes techniques

Nous pouvons choisir entre deux systèmes d'écoutes : les haut-parleurs ou le casque.

Dans le cadre d'un système de réalité virtuelle tel que le CAVE, les haut-parleurs posent trois problèmes majeurs :

- placés devant les écrans, ils nuiraient à la vision de l'auditeur.
- placés derrière, les écrans nuiraient à l'audition.
- enfin, l'acoustique du lieu d'écoute peut être gênant si nous ne sommes pas en environnement anéchoïque, ce qui est le cas du CAVE.

La solution technique retenue est donc celle de l'écoute au casque, on parle alors de système binaural. Le choix du casque permet de s'affranchir des trois problèmes cités plus haut, en effet il offre à l'auditeur une qualité d'écoute optimale, indépendamment de l'acoustique de la pièce dans laquelle l'expérience se déroule, ceci sans gêner sa vision.

Entre un émetteur et un récepteur de type casque, il existe trois liaisons possibles : filaire, infrarouge ou haute-fréquence.

Une liaison filaire risquerait de provoquer des gênes chez l'auditeur (sensorielles, visuelles, limitation des déplacements). Une liaison infrarouge perturberait et serait perturbée par les caméras de tracking *ART* permettant de collecter les coordonnées de l'auditeur. La solution retenue est donc celle du casque sans fil en liaison HF.

## Chapitre 4

# Eléments de perception acoustique

Nous avons vu que la spatialisation d'une source sonore est rendue possible grâce à Spat, l'imagerie du système de réalité virtuelle, quand à elle, est commandé par Virtools. Le but de notre application est de faire communiquer les deux logiciels, mais dans un seul sens : récupération des informations sources/auditeur fournies par Virtools, traitement puis propagation de ces données dans Spat, ceci en temps réel. Il est donc important de bien comprendre quels sont les principaux paramètres d'entrées (et leurs rôles) d'un logiciel de spatialisation comme Spat, d'une manière plus générale, comprendre comment notre système auditif permet de nous donner des informations sur la position et la vitesse d'une source sonore.

Dans l'espace, on localise une source sonore unique et immobile par 3 paramètres :

- son azimut (angle dans le plan horizontal).
- son élévation (angle dans le plan vertical).
- la distance à laquelle elle se trouve par rapport à l'auditeur.

ou bien

- son ordonnée.
- son abscisse.
- sa hauteur.

Notre cerveau évaluent ces paramètres aux moyens d'*indices acoustiques de localisation*.

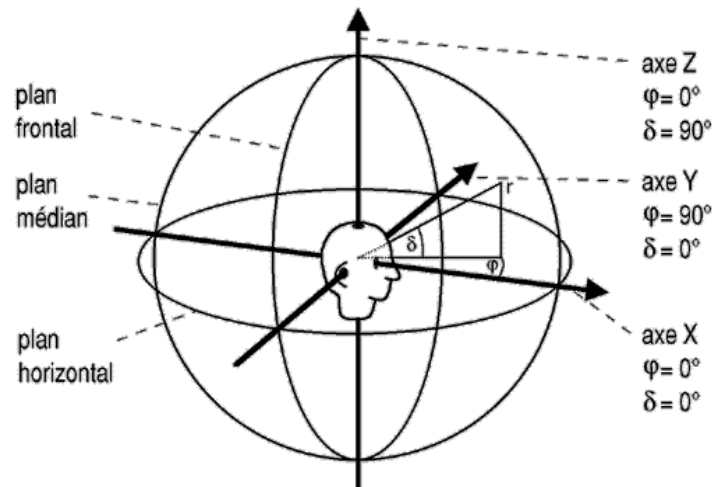


FIG. 4.1 – Système de représentation sphérique de l'espace par rapport à la tête où  $r$  est la distance,  $\delta$  est l'élévation et  $\varphi$  est l'azimut [d'après Blauert]

## 4.1 Les indices acoustiques de localisation

Ce chapitre extrait les points essentiels de la thèse de Véronique Larcher [4], de la thèse de Sylvain Busson [7] et de l'article de Daniel Courville [5].

### Les indices interauraux

En 1907, Lord Rayleigh publie la *théorie duplex* basée sur la définition de deux différences interaurales permettant de décrire les mécanismes simples de la perception en azimut :

- la différence interaurale de temps (ou ITD pour *Interaural Time Difference*).
- la différence interaurale d'intensité (ou ILD pour *Interaural Level Difference*).

### La différence interaurale de temps

Il s'agit de la différence de temps que met l'onde sonore pour parcourir la distance d'une oreille à l'autre, en contournant la tête. Sa contribution est prédominante en basses fréquences ( $f < 1500$  Hz). Cette différence temporelle varie selon l'azimut de la source sonore, elle est nulle pour une source située à  $0^\circ$  ou  $180^\circ$  d'azimut (devant ou derrière l'auditeur), et maximale pour  $90^\circ$  ou

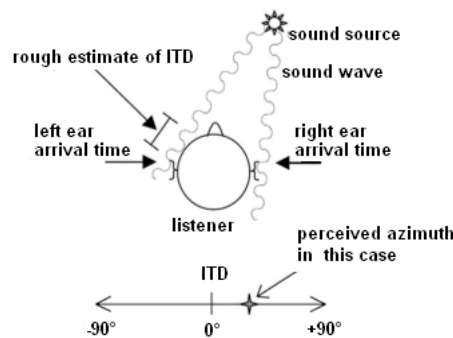


FIG. 4.2 – Utilisation de l'ITD pour la localisation en azimuth. [d'après Cheng and Wakefield (2001)]

270° d'azimut (gauche ou droite, voir figure 4.1). La localisation en azimuth est une fonction quasi-linéaire de l'ITD. Une source est en général plus proche de l'oreille recevant le front d'onde en premier. Plus l'ITD est grand et plus la source paraît latéralisée (voir figure 4.2). Une approximation de la différence interaurale de temps en fonction de l'azimut  $\theta$  est donnée par la formule de Woodworth (1962) :

$$ITD_{sphere}(\theta) = \frac{a}{c}(\sin(\theta) + \theta)$$

où  $a$  est la distance entre deux oreilles (environ 21,5 cm) et  $c$  la vitesse de propagation du son.

### La différence interaurale d'intensité

Il s'agit de la différence d'amplitude de l'onde sonore entre les deux oreilles, cet indice de localisation est complémentaire de l'ITD, sa contribution majeure se situe en haute fréquences ( $f > 1500$  Hz). Une source est perçue plus proche de l'oreille recevant le plus d'énergie acoustique. Plus l'ILD est important, plus la source paraît latéralisée. L'ILD est calculée par le rapport de l'énergie entre le signal présenté à l'oreille droite et celui présenté à l'oreille gauche.

Selon les expériences de Véronique Larcher dans [4], pour les positions situées à l'arrière de la tête, au dessus du plan horizontal ( $100^\circ < \text{élévation} < 180^\circ$ ), la différence interaurale d'intensité mesurée est nettement inférieure à celle observée pour les positions frontales, cette différence constitue donc un indice prépondérant pour la discrimination avant-arrière.

Dans l'étude citée précédemment, on considérait que la source était en champ lointain ( $d > 1\text{m}$ ). En 1999, Brungart et Rabinovitz modélisent les différences interaurales d'intensité et de temps en champ proche ( $d < 50\text{cm}$ ) et obtiennent le résultat suivant : la différence interaurale d'intensité augmente fortement quand  $d$  diminue alors que la différence interaurale de temps varie de moins de 12% entre les valeurs établies pour le champ lointain et pour le champ proche. Ils montrent ainsi que la différence interaurale d'intensité, en plus d'être un indice de discrimination avant arrière, est aussi un indice de distance de la source.

### **Les indices monauraux**

Il existe des positions pour lesquelles les différences interaurales sont identiques. L'ITD et l'ILD ne peuvent donc pas suffire à déterminer l'incidence d'une source sonore. Pour résoudre cette indétermination, le système auditif a recours à des indices dits "monauraux". Comme pour les indices interauraux, le système auditif interprète l'information monaurale en pratiquant conjointement une analyse dans le domaine temporel et une analyse dans le domaine fréquentiel. Ces indices sont propres à la forme de notre oreille externe, ainsi, pour les fréquences supérieures à 2000 Hz, chaque variation angulaire d'une source sonore dans le plan médian (aucune différence interaurale de temps ou d'amplitude, voir figure 4.1) résulte en une modification par les creux et les bosses du pavillon de l'oreille de la réponse en fréquence.

Un modèle, proposé par Theile en 1986, suppose que nous possédons une base de données personnelle d'indices de localisation, acquise et mémorisée au cours de notre vie. Ce modèle montre qu'une bonne localisation est liée à la connaissance préalable du timbre de la source. En effet Blauert, en 1997, a observé que de meilleures performances de localisation sont obtenues lorsque le stimulus est une voix familière plutôt qu'une voix inconnue. De plus, puisque l'information utilisée est d'ordre spectral uniquement, un son de type bruit blanc sera plus facile à localiser.

### **Conclusion**

Les indices de localisation permettent à notre système auditif de localiser correctement une source. Ils sont de deux natures (indices interauraux, indices monauraux) et possèdent dans les deux cas des attributs fréquentiels et temporels variant avec l'incidence. En réalité, d'autres facteurs interviennent comme la conductivité osseuse du corps, ou les mouvements de la tête qui nous permettent de diminuer l'incertitude quand à la localisation d'une source, mais en assimilant les indices monauraux et interauraux à des filtres linéaires dont la fonction de transfert dépend de la position de

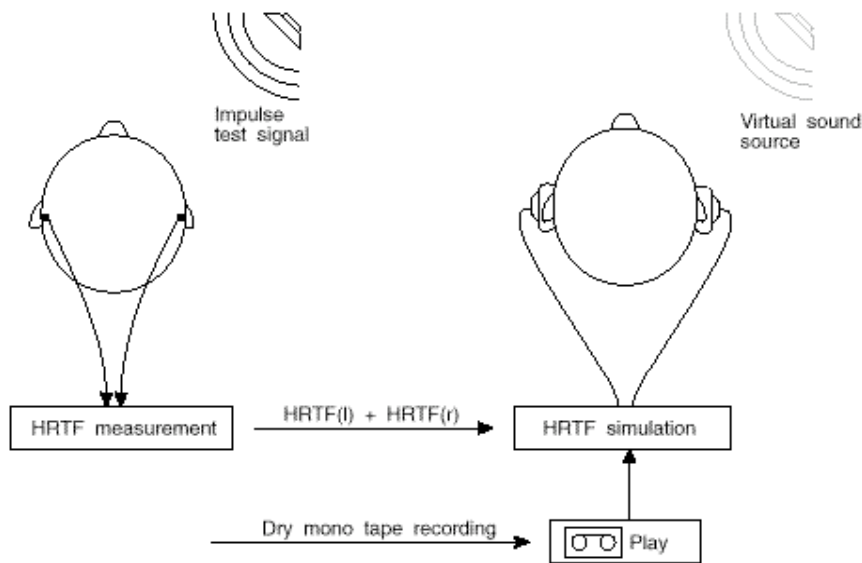


FIG. 4.3 – Mesure et simulation d’HRTF

la source, on arrive à reproduire une sensation de spatialisation, c’est ce que font les spatialisateurs comme Spat en utilisant, de manière plus générale, les HRTF.

## 4.2 Les HRTF

Les modifications spectrales que subit une source sonore pour arriver aux oreilles d’un auditeur constituent les filtres liés à la tête, ou HRTF pour *Head Related Transfer Functions*. Les HRTF décrivent la manière dont un son donné (paramétré en fréquence et en position) est filtré par les diffractions et réflexions de notre corps, principalement la tête, le pavillon et le torse avant qu’il n’atteigne le tympan et l’oreille interne. Les HRTF contiennent toutes les informations de spatialisation, l’ITD et l’ILD peuvent être extraits des HRTF. Pour reproduire la sensation d’une source sonore provenant d’une incidence donnée, il suffit donc de filtrer un son par les HRTF droite et gauche mesurées pour cette position. C’est le principe de la synthèse binaurale.

D’après le modèle proposé par Mehrgardt et Mellert en 1977, chaque HRTF peut être représenté sous forme d’un retard et d’un spectre d’amplitude, mais les HRTF peuvent difficilement être exprimées à l’aide d’une formule, car elles dépendent de la position du son, de sa fréquence et de la morphologie de l’auditeur, autrement dit, à chaque individu correspond une paire d’HRTF différente. Ainsi, une écoute binaurale proposant à l’auditeur des HRTF différentes des siennes génère des artefacts de localisation (ambiguïté

avant-arrière). Dans l'idéal il faudrait donc connaître les HRTF de chaque auditeur, mais le protocole permettant de les mesurer est assez lourd : on installe l'auditeur en chambre anéchoïque, on place un micro dans chacun de ses tympans et on mesure les fonctions de transferts des oreilles droite et gauche pour 300 positions de source au moins (d'après Blauert), les HRTF liées aux autres positions étant déterminées par interpolation. L'Ircam possède une banque de données d'HRTF : ces mesures ont été faites sur 51 individus et un mannequin de type KEMAR en envoyant un signal sonore tous les 5° degrés en azimut et tous les 10° en élévation, ceci sur une sphère d'un rayon de 3 mètres dont le centre est la tête de l'auditeur. Ainsi Spat propose 52 HRTF qu'il faut essayer une à une pour trouver laquelle nous correspond le mieux.

Nous avons développé, en parallèle de ViSpat, un logiciel permettant à un auditeur de déterminer les HRTF qui lui correspondent le mieux parmi celles proposées par Spat. Dans cette application, l'auditeur est représenté à l'écran par un point au centre d'un cercle symbolisant le plan horizontal. Quatre sources (bruits blancs) placés à quatre azimuts différents sont déclenchés l'une après l'autre, pour chaque source, l'utilisateur clique sur l'écran à l'endroit où il pense l'avoir localisé. On calcul un écart moyen sur les quatre sources entre la réalité et les réponses de l'auditeur. Cet écart moyen est mémorisé et l'expérience est répétée pour une paire d'HRTF différente. Ce que l'utilisateur ne sait pas, c'est que les quatre sources sont toujours placées au même endroit quels que soient les HRTF choisies, ainsi, l'écart moyen le plus petit correspond à l'HRTF la mieux adaptée à l'auditeur.

Pour plusieurs sujets testés, on s'aperçoit que :

- aucune HRTF n'est vraiment prédominante, chaque individu trouve une HRTF différente des autres.
- pour un même individu, il est fréquent que des écarts de plus de 45° soient constatés d'une HRTF à une autre bien que les sources soient localisées au même endroit.

Ceci justifie l'importance de déterminer quelle HRTF est la plus à même de convenir à l'auditeur. En pratique l'HRTF moyenne mesurée avec le mannequin KEMAR convient assez bien compte-tenu du couplage vision/audition, c'est-à-dire que dans le cas d'une réalité virtuelle, la vision de la source compense une éventuelle imprécision de localisation sonore.

## Conclusion

Toutes les informations de localisation sont contenues dans les HRTF, la synthèse binaurale consiste alors à filtrer un son vierge (non réverbéré) par l'une de ces empreintes spatiales. Mais l'inconvénient majeure est que ces HRTF sont individuelles, et déterminer les HRTF d'un individu est inenvisageable dans notre cas, en pratique, on choisit donc un jeu d'HRTF moyennes résultants de mesures sur la marionnette KEMAR. Reste à rendre compte de la sensation de mouvement, pour ce faire, Spat intègre un indice acoustique dynamique, plus communément appelé effet Doppler.

### 4.3 Simulation du mouvement : effet Doppler

L'effet Doppler est le décalage entre la fréquence de l'onde émise et de l'onde reçue lorsque l'émetteur et le récepteur sont en mouvement l'un par rapport à l'autre.

Supposons que l'émetteur et le récepteur se déplacent sur une droite munie d'un référentiel galiléen. Il y a trois référentiels à considérer :

- (1) : le référentiel du milieu dans lequel se propage l'onde. On note  $v$  la vitesse de l'onde dans ce référentiel.
- (2) : le référentiel lié à l'émetteur (source) : appelons  $v_s$  la vitesse algébrique de l'émetteur par rapport au référentiel (1).
- (3) : le référentiel lié au récepteur : appelons  $v_r$  la vitesse du récepteur par rapport au référentiel (1).

Si  $f_0$  est la fréquence de l'onde dans le référentiel de la source, alors le récepteur va recevoir une onde de fréquence  $f_a$  (fréquence apparente).

$$f_a = \frac{v-v_r}{v-v_s} \cdot f_0$$

En effet, supposons que la source émette des bips à une fréquence  $f_0$  et que le mouvement relatif entre l'émetteur et le récepteur se fasse selon la droite les joignant. Lorsque le deuxième bip est produit, le premier bip a parcouru une distance  $d_0$  :

$$d_0 = v \cdot f_0$$

Dans le référentiel (1), avec  $T_0 = \frac{1}{f_0}$ , la source s'étant déplacée de  $v_s \cdot T_0$  pendant le temps  $T_0$ , la distance séparant deux bips est :

$$d_1 = (v - v_s) \cdot T_0$$



Calculons le temps  $T_a$  séparant la réception des deux bips par le récepteur. Ce dernier reçoit le premier bip. Au bout de ce temps  $T_a$ , il a parcouru la distance  $v_r \cdot T_a$  au moment où il reçoit le deuxième bip. Ce deuxième bip aura donc parcouru la distance

$$d_2 = d_1 + v_r \cdot T_a = v \cdot T_a$$

Ce qui donne bien :

$$f_a = \frac{1}{T_a} = \frac{v-v_r}{d_1} = \frac{v-v_r}{v-v_s} \cdot \frac{1}{T_0} = \frac{v-v_r}{v-v_s} \cdot f_0$$

Si seule la source est mobile par rapport au référentiel ( $v_r = 0$ ), on a alors :

$$f_a = \frac{v}{v-v_s} \cdot f_0$$

Et si seul le récepteur est mobile par rapport au référentiel ( $v_s = 0$ ), on a :

$$f_a = \frac{v-v_r}{v} \cdot f_0$$

On voit clairement que les deux situations ne sont pas symétriques : en effet, si le récepteur fuit l'émetteur à une vitesse supérieure à  $v$ , il ne recevra jamais d'onde, alors que si l'émetteur fuit un récepteur immobile, celui-ci recevra toujours une onde. On ne peut pas inverser le rôle de l'émetteur et du récepteur. Dans le cas classique, il y a dissymétrie dans le décalage fréquentiel selon que l'émetteur ou le récepteur est en mouvement (les fréquences reçues diffèrent par les termes du second ordre pour une même fréquence d'émission). Cette dissymétrie est due à la présence du milieu dans lequel se propagent les ondes, elle est justifiée pour les ondes sonores.

## Conclusion

L'effet Doppler est un indice acoustique dynamique. Son caractère dissymétrique permet à notre système auditif de savoir si une source s'éloigne ou se rapproche de nous. En fait il nous permet de connaître la sens de déplacement de la source, mais qu'en est-il de la distance exacte qui nous sépare d'elle à l'instant  $t$  ? Le chapitre suivant permet d'expliquer ce qui permet à notre système auditif d'être relativement précis sur la distance qui nous sépare d'une source, ou de comparer la distance d'une source par rapport à une autre.

## 4.4 Impression de distance

Ce chapitre présente les points essentiels concernant les caractéristiques responsables de la localisation d'une source sonore du point de vue de la distance. Les études qui vont suivre proviennent d'un rapport rédigé par Georges Canevet [8].

L'être humain évalue difficilement l'éloignement d'une source sonore excepté pour des sources connues ou familières. Ceci dit, trois principales caractéristiques du champ acoustique peuvent recréer la sensation d'éloignement : les variations d'intensité, le rapport du son direct au son réverbéré et certaines modifications spectrales typiques.

### L'amplitude

Il est évident que lorsqu'une source d'énergie constante s'éloigne d'un auditeur, l'intensité diminue aux oreilles de cet auditeur et inversement lorsque cette source s'en rapproche. C'est en 1969 que Gardner met en évidence la corrélation des variations de distance et des variations d'intensité. L'expérience consistait à placer un auditeur devant cinq haut-parleurs plus ou moins éloignés et de lui demander, pour chaque son entendu, de quel haut-parleur provenait le son. A l'insu de l'auditeur, seuls deux haut-parleurs étaient opérationnels. Pour chaque auditeur interrogé, les réponses étaient à peu près les mêmes et mettaient en scène les 5 haut-parleurs. L'expérience montre alors que le jugement de distance dépend des niveaux d'émission acoustique mais n'a rien à voir avec la distance réelle.

### La réverbération

Voici un constat de la vie quotidienne, lorsque dans une salle un locuteur s'éloigne, le rapport du son direct au son réverbéré décroît. Cette décroissance sera associée par l'auditeur à un éloignement et deviendra un indice pour l'évaluation de la distance. Mershon et King, en 1975 ont montré, en utilisant des bouffées de bruit blanc présentées par des hauts parleurs dans un tunnel (6m de long, 1,8m de large et 2,5m de haut) à parois réfléchissantes, que la réverbération a une qualité d'indice absolu, c'est-à-dire qu'elle permet le positionnement d'une source dans l'espace proche de l'individu. Contrairement à l'intensité qui elle ne permet de détecter que des variations de distance.

## Les variations spectrales

Enfin, un signal sonore voit ses caractéristiques spectrales varier avec la distance à cause de l'absorption inégale entre les graves et les aigües. Ainsi un sujet est capable d'évaluer les distances relatives d'émission pour un signal donné qu'il connaît. Butler, en 1980, a enregistré des bouffées de bruit (de 30 ms) à spectre passe-haut, passe-bas et large bande en chambre sourde puis en salle normale. Ces enregistrements ont été présentés à 16 sujets, qui devaient indiquer la direction et la distance apparente de la source. L'expérience montre que la composition spectrale des signaux joue un rôle déterminant sur la distance apparente. Les bruits filtrés passe-bas semblent plus éloignés que les bruits filtrés passe-haut.

## Conclusion

L'évaluation de la distance d'un son est rendue précise essentiellement grâce à son amplitude, sa réverbération et ses variations spectrales. Ce sont sur ces paramètres impersonnels, associés à l'indice interaural d'intensité, que repose Spat, et qui permet de retrouver cette sensation de sons "spatialisés".

## 4.5 Conclusion sur les éléments de perception acoustique

La simulation de l'espace sonore sous écouteurs se traduit principalement par un filtrage (HRTF). La difficulté est que ces filtres nous sont individuels, ainsi 2 personnes différentes écoutant un même son spatialisé n'auront pas forcément la sensation de l'entendre au même endroit. Ainsi, le logiciel de spatialisation Spat peut filtrer les sons en entrées par plusieurs paires de HRTF différentes afin de choisir celle qui correspond le mieux à l'auditeur... encore faut-il avoir le temps de toutes les tester. Dans notre projet on utilisera par défaut des HRTF dites "moyennes". Ce choix se justifie par le fait que nous voyons les sources que nous entendons, ainsi tout écart de localisation sonore est rapidement corrigé par notre cerveau lorsqu'il couple les données sonores aux données visuelles, ainsi après quelques minutes d'immersion, nous serons tout à fait capable de nous localiser grâce aux "oreilles d'un autre". Mais le filtrage par HRTF ne rend pas compte de tous les éléments de perception acoustique comme le mouvement, ou la distance sur lequel ce procédé reste imprécis. Ainsi Spat agit sur d'autres paramètres comme l'amplitude du signal sonore, la réverbération de la pièce, l'effet Doppler ou encore l'absorption de l'air.

## Chapitre 5

# Présentation de ViSpat

Les principes de la spatialisation sonore ayant été énoncés, la suite de ce rapport portera sur le travail réalisé lors de ce stage, c'est-à-dire le développement d'un logiciel adapté aux utilisations du CRVM de Luminy, sorte de couche supérieure à Spat. Par souci de commodité, dans ce rapport le logiciel sera baptisé *ViSpat* (symbolisant ainsi l'union entre réalité virtuelle et spatialisation sonore).

Le schéma 5.1 représente le fonctionnement général de ViSpat. Toutes les informations concernant les sources sonores et l'auditeur (positions, orientations, directivité, gain etc.) sont connues et/ou contrôlées par le CRVM, elles sont envoyées à ViSpat en temps réel par trames UDP. Après traitement, les paramètres de ces trames sont envoyés à Spat. Une interface utilisateur permet de visualiser les erreurs de transmission et de contrôler certains paramètres (choix des HRTF, de la réverbération, arrêt du processus etc.).

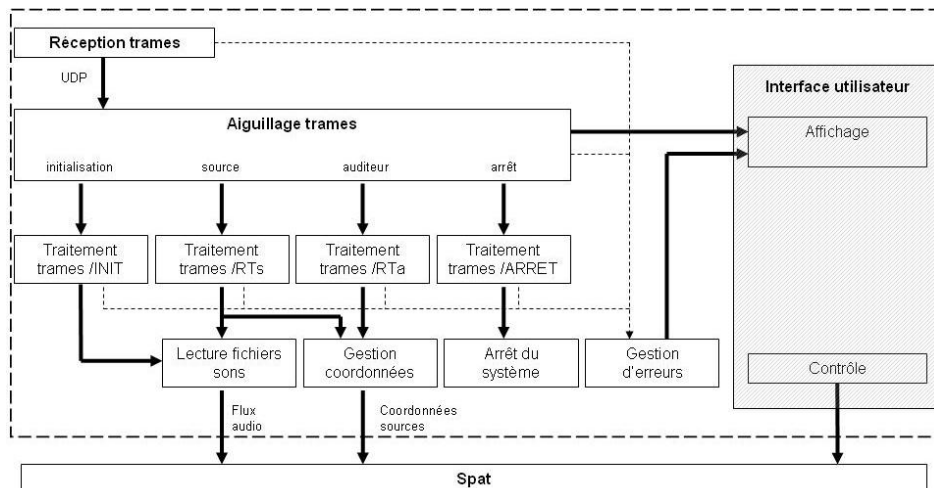


FIG. 5.1 – Fonctionnement général de ViSpat

## 5.1 Protocole de communication

Une première réunion a eu lieu concernant la communication entre le système informatique du CRVM et l'ordinateur du LMA sur lequel ViSpat a été développé, la problématique étant la suivante : dans le CAVE, les coordonnées de l'utilisateur sont collectées par un système de tracking *ART* basé sur quatre caméras infrarouges. C'est grâce aux informations délivrées par ces caméras que s'effectue la mise à jour des scènes visuelles en temps réel (à 60 Hz). L'idée est de récupérer ces mêmes informations pour la mise à jour des scènes sonores.

Les problèmes qui se posent à établir un protocole de communication sont multiples : comment respecter le temps réel ? Quel type de communication choisir ? Sous quel format transmettre les informations ? etc. Voici les réponses apportés et les principaux points de discussion lors de cette réunion.

### Les informations à transmettre

Pour un système de spatialisation sonore, il est nécessaire de pouvoir communiquer :

- les informations concernant l'auditeur
  - sa position (x, y, z)
  - son orientation (3 angles d'Euler)
- les informations concernant chaque source sonore
  - sa position
  - son orientation
  - son niveau sonore de la source (relativement aux autres sources)
  - sa directivité
  - son état (à l'arrêt ou en lecture)

### Le protocole d'échange de données

Nous avons choisi de communiquer ces informations par Ethernet en suivant le protocole UDP. Ces informations se présenteront donc sous forme de trames. Pour être compatible avec MAX/MSP, le format de ces trames doit suivre le protocole OSC (couche supérieure à UDP) dont les caractéristiques sont explicitées sur le site [4].

## Les types de trames

ViSpat peut gérer 6 types de trames :

- une trame d'initialisation de l'expérience
- les trames relatives aux sources
- les trames relatives à l'auditeur
- la trame de gain général
- les trames de réverbération
- une trame d'arrêt de l'expérience

### La trame d'initialisation

Son rôle est de donner au système les informations concernant les sources, valables durant toute la durée de l'expérience. Elle informe le système sur :

- le nombre total de sources
- le numéro attribué à chaque source
- la méthode de synthèse utilisée (propre à chaque son).

La suite de la trame dépend de la méthode de synthèse utilisée. Pour la méthode dite de *sampling* (la seule développée à ce jour, voir paragraphe *Méthode de synthèse* page 31), la trame informe le système sur :

- le chemin d'accès au fichier son
- la lecture en boucle ou non de chaque son
- la directivité de chaque son

Chaque son est chargé dans un buffer ayant comme paramètre d'entrée le chemin d'accès au fichier. Il est ensuite joué dès réception d'un signal de lecture (défini dans la trame source, voir 7.1.1 page 44). C'est pour l'instant la première et la seule méthode de synthèse utilisée, mais il n'est pas à exclure que d'autres méthodes viennent se greffer au système, c'est pour cela que l'on attribue un numéro spécialement dédié au choix de la méthode de synthèse.

D'autres paramètres pourront à l'avenir compléter cette trame comme l'application ou non de l'effet Doppler sur la source, la prise en compte ou non de l'absorption de l'air etc. Aujourd'hui, ces paramètres doivent être rentrés manuellement pour chaque source à l'aide d'une interface graphique avant ou en cours d'expérience.

### **Les trames sources**

Leur rôle est de donner au système les informations concernant les sources au cours de l'expérience. A une source correspond une trame source repérable par son numéro (trame source 0 pour la source 0, trame source 1 pour la source 1 etc. . . ). Elles sont envoyées à une fréquence d'environ 60 Hz et sont traitées si modification d'un ou de plusieurs de leurs paramètres. Un trame source informe le système sur :

- le numéro de la source concernée
- son niveau sonore en dB
- sa position cartésienne par rapport au CAVE
- son orientation par rapport au CAVE
- son état : lecture ou arrêt

### **La trame auditeur**

Son rôle est de donner au système les informations concernant l'auditeur au fur et à mesure de son évolution dans l'espace. Ces informations sont :

- sa position cartésienne par rapport au CAVE
- son orientation par rapport au CAVE

### **La trame réverbération**

Cette trame permet de modifier la réverbération générale du paysage sonore. Selon l'endroit ou l'utilisateur se trouve, on envoie un entier correspondant un certain paramétrage de réverbération.

### **La trame de gain**

Il s'agit d'un entier (en dB) qui agit sur toutes les sources sonores à la fois et modifie ainsi le gain général de la scène.

### **La trame arrêt**

Dès réception de la trame arrêt le système coupe tous les flux audio et les buffers sont vidés.

La constitution de toutes ces trames ainsi que leur traitement est développée plus précisément en annexe au chapitre 7.1.1 à la page 41.

## **Méthode de synthèse**

ViSpat gère la synthèse et la spatialisation des sons. Dans cette première version, la synthèse consiste à lire des sons préenregistrés à une certaine fréquence d'échantillonnage (méthode de *sampling*). Chaque monde (décor virtuel) doit avoir au préalable sa bibliothèque de sons associée et enregistrée dans un dossier fixe de la machine.

## **Gestion d'erreurs**

Un système de gestion d'erreur permet de détecter quelques causes classiques de mauvais fonctionnement du logiciel (section 5.6 page 36).

## **Autonomie**

Le système est autonome : au démarrage de l'ordinateur, ViSpat s'ouvre automatiquement et est prêt à recevoir la trame d'initialisation, pas besoin de manipuler le logiciel.

Les informations traitées sont les paramètres d'entrées de Spat.

## **Documentation logiciel**

Une formation accompagnée d'une fiche d'utilisation récapitulative à été donnée au personnel du CRVM afin de maîtriser toute la partie audio (nouvelle pour eux). Voir annexe chapitre 7.31 page 75.

## **5.2 La gestion des coordonnées**

Comme nous l'avons vu plus haut, il y a deux types de coordonnées dans les trames sources et auditeur :

- les positions
- les orientations

Les positions sont gérées par le patch *ConvCooPos* : Annexe 7.14 page 62 et les orientations par le patch *YawPitch* : Annexe 7.7 page 55.



Le patch ConvCooPos contient un objet ViRotation, qui a été implémenté spécifiquement pour ce projet en langage C afin d'optimiser les temps de calcul, voir 7.15 page 62. ViRotation applique un changement de repère sur les sources afin de les obtenir dans le repère de l'auditeur et non plus du CAVE. Il convertit ensuite toutes les coordonnées cartésiennes en coordonnées sphériques (Spat travaille en sphérique). Enfin toute nouvelle position n'est propagée uniquement pour des variations supérieures à un seuil fixé : inutile que Spat relance une série de calculs engendrant des variations sonores imperceptibles par nos oreilles. Dans l'exemple de la page 62, *virotation 1 10* signifie que la nouvelle coordonnée de la source ne sera transmise à Spat que pour une variation en position supérieure à 1 mètre ou une variation en rotation supérieure à 10°, ce qui est suffisant pour notre projet.

### Gestion des positions

La position de l'auditeur est repérée par le système de tracking sous forme de 3 entiers correspondants à sa position cartésienne (x, y, z) dans le repère du CAVE en mm. Ces coordonnées sont ensuite fournies au système de spatialisation via une trame de type /RTa.

Pour Spat, l'auditeur est toujours à l'origine du repère, ce qui n'est pas le cas dans le CAVE qui à sa propre origine, donc les coordonnées de la source sont recalculées en permanence en fonction des déplacements de l'auditeur, celles-ci peuvent également évoluer sans que l'auditeur ne bouge.

Dans un premier temps les coordonnées de la source subissent un changement de repère afin de les exprimer dans le repère de l'auditeur et non plus du CAVE :

notations :

*SA* : source dans le repère de l'auditeur.

*SC* : source dans le repère du CAVE.

*AC* : auditeur dans le repère du CAVE.

calculs :

$$x_{SA} = x_{SC} - x_{AC}$$

$$y_{SA} = y_{SC} - y_{AC}$$

$$z_{SA} = z_{SC} - z_{AC}$$

## Conversion repère cartésien vers repère sphérique

Pour communiquer avec Spat, il est nécessaire de lui apporter les informations concernant les sources, en coordonnées sphériques. Les filtres agissant sur les sources sont directement calculés en fonction de la distance, de l'élévation et de l'azimut de la source par rapport à l'auditeur.

Il faut exprimer  $d$ , la distance entre l'auditeur et la source,  $\theta$  et  $\Gamma$  respectivement l'azimut et l'élévation de la source par rapport à l'auditeur en fonction des coordonnées cartésiennes de la source par rapport à l'auditeur  $X_{SA}$ ,  $Y_{SA}$  et  $Z_{SA}$ .

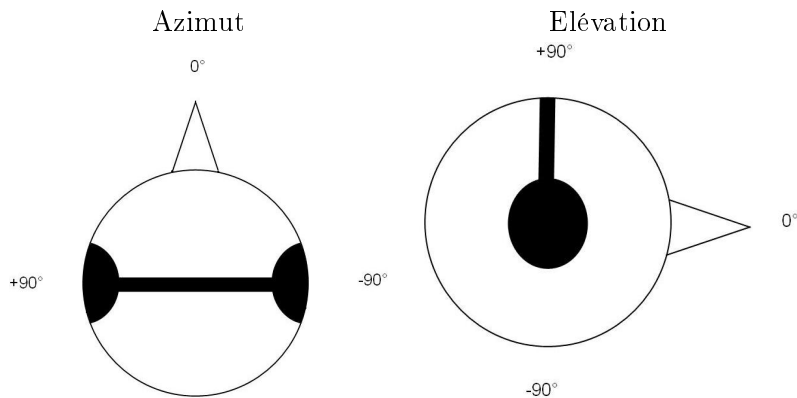
D'après le théorème de Pythagore, on a :

$$d = \sqrt{X_{SA}^2 + Y_{SA}^2 + Z_{SA}^2}$$

Puis d'après les formules de trigonométrie, on a :

$$\begin{aligned}\theta &= \text{Arctan}\left(\frac{Y_{SA}}{X_{SA}}\right) \text{ si } Y_{SA} > 0 \\ \theta &= -\text{Arctan}\left(\frac{Y_{SA}}{X_{SA}}\right) \text{ si } Y_{SA} < 0 \\ \Gamma &= \text{Arcsin}\left(\frac{Z_{SA}}{d}\right)\end{aligned}$$

Ces données sont transmises à Spat qui gère l'azimut et l'élévation de la manière suivante :



### 5.3 Gestion de l'orientation des sources dans le plan horizontal

Il est possible dans Spat de définir, pour chaque source, un angle d'ouverture (ou directivité)  $\gamma$  et une orientation. L'orientation d'une source se définit par deux angles, un angle dans le plan horizontal  $\theta_z$  et un angle dans le plan vertical  $\theta_x$ .

Dans Spat, l'angle de directivité de la source est défini par rapport à l'auditeur et non par rapport à un repère fixe (en fait, Spat ne prévoit pas que l'auditeur soit mobile). Ce qui a pour conséquence que la source montre toujours la même face à l'auditeur même si celui-ci tourne autour d'elle.

Il a donc fallu implémenter un asservissement permettant de simuler une directivité fixe quel que soit les mouvements de l'auditeur : voir figure 7.4 page 47. *Le détail des calculs est en annexe, chapitre 7.1.3 page 47.*

### 5.4 Gestion de l'azimut et de l'élévation

Pour localiser une source dans le plan horizontal et vertical, Spat demande deux valeurs : l'azimut et l'élévation (coordonnées sphériques).

Ainsi lorsqu'un auditeur écoute une source sur sa droite et à la hauteur de son oreille, soit un azimut de  $-90^\circ$  et une élévation de  $0^\circ$ , et que cet auditeur fait un demi-tour sur lui-même, la source doit alors passer sur l'oreille gauche, c'est-à-dire (en langage Spat) que l'azimut de la source doit basculer à  $+90^\circ$  (l'élévation vaut toujours  $0^\circ$ ). En théorie, il suffit donc de récupérer l'angle de rotation sur l'axe de la hauteur, ici  $180^\circ$ , et de l'ajouter à l'ancien azimut ( $-90^\circ$ ) pour trouver le nouvel azimut ( $+90^\circ$ ). Seulement Virtools, le moteur 3D utilisé par le CRVM qui nous envoie les angles de rotations de l'auditeur, ne fonctionne pas tout à fait comme ça.

Premièrement, Virtools fonctionne en repère cartésien contrairement à Spat qui fonctionne en repère sphérique. Ensuite, Virtools utilise  $y$  comme axe de la hauteur (et non pas  $z$ ). Mais la subtilité principale de Virtools est la suivante : pour toute rotation de l'auditeur sur l'axe de la hauteur supérieure à  $90^\circ$ , il effectue en réalité deux rotations de  $180^\circ$  sur les deux autres axes, puis termine avec une rotation d'un angle complémentaire à  $180^\circ$  sur l'axe de la hauteur.

Prenons un exemple : tourner sur soi d'un angle de  $100^\circ$  n'est pas traduit par Virtools comme une simple rotation de  $100^\circ$  sur l'axe des y (hauteur dans Virtools) mais par tourner de  $180^\circ$  sur l'axe des x, puis de  $180^\circ$  sur l'axe des z, et enfin de  $180-100=80^\circ$  sur l'axe des y. Ce qui visuellement reviens au même. Mais ces changements brutaux sur les axes z et x de 0 à  $180^\circ$  provoquent des "à-coups" sonores lorsque ces angles servent uniquement à calculer l'élévation. Avec la logique Virtools, tourner sur soi modifie donc l'élévation, ce qui n'est pas intuitif.

Dans l'algorithme il faut donc tenir compte en permanence de tous les angles même si physiquement un seul d'entre eux est modifié. *Le détail des calculs est en annexe, chapitre 7.1.4 page 48 .*

## 5.5 L'interface utilisateur

L'interface permet à l'utilisateur de visualiser les informations sur les sources et de contrôler certains paramètres audio ou de spatialisation. Voir Annexes 7.2 page 52.

### L'affichage des messages d'erreurs

Ce cadre permet de savoir si une trame ne suit pas le protocole établi. Il informe sur le type et la cause de l'erreur. Dans le cas d'une erreur sur la trame source, le message d'erreur indique quel numéro de source est en cause.

### Informations sur les sources

Pour chaque source sonore, on peut visualiser

- le nom du fichier son correspondant à l'auditeur 0, celui correspondant à l'auditeur 1 etc.
- si celle-ci sera lue en boucle ou pas
- si le buffer que cette source occupe est correctement rempli
- son état : play ou stop

### L'utilisation CPU

La spatialisation sonore en temps réel demande beaucoup de puissance de calcul dès lors que plusieurs sources sont traités simultanément, un indicateur indique sous forme de flottant le pourcentage d'utilisation du CPU afin de prévenir une surexploitation du système et un éventuel bug.

## Choix des HRTF

Comme expliqué dans le paragraphe 4.2 page 21, il est important d'utiliser les HRTF les mieux adaptés à l'auditeur. Pour l'aider dans ce choix, un logiciel de préparation à été implémenté. Il s'agit d'un test d'environ une heure, permettant de déterminer, parmi les HRTF proposées, celles qui nous correspond le mieux.

## L'ajustement général du volume

Un slider permet d'ajuster le niveau général de l'environnement sonore.

## L'arrêt du système

L'arrêt du système peut être provoqué en cliquant sur le bouton ARRÊT, son effet sera le même que la réception d'une trame arrêt (voir plus haut, la trame d'arrêt).

## 5.6 La gestion d'erreurs

ViSpat permet de signaler à l'utilisateur, via l'affichage d'un message sur l'interface, certaines erreurs dues à une mauvaise construction des trames.

Les erreurs gérées dans cette première version sont :

- un mauvais numéro de méthode de synthèse dans la trame /INIT. Voir patch Error100 : Annexe 7.23 page 70
- une déclaration de source mauvaise ou inexistante dans la trame /INIT. Voir patch Error200 : Annexe 7.24 page 71
- un nombre de source invalide dans la trame /INIT. Voir patch Error400 : Annexe 7.25 page 71
- numéro de sources ou nombre de paramètres renseignés dans la trame /RTs incorrects. Voir patch Error500 : Annexe 7.26 page 72
- nombre de paramètres renseignés dans la trame /RTa incorrects. Voir patch Error600 : Annexe 7.27 page 73
- type de trame inconnu par le système d'aiguillage. Voir patch Error700 : Annexe 7.28 page 73
- format du fichier son différent de .wav ou .mp3. Voir patch Error800 : Annexe 7.29 page 74
- booléen informatif sur la lecture en boucle incorrect ou inexistant dans la trame /INIT. Voir patch Error900 : Annexe 7.30 page 74

## 5.7 La lecture des fichiers sons

La lecture des fichiers sons est gérée par le patch LecS : Annexe 7.20 page 68, elle nécessite 4 informations :

- quelle fichier son lire (buffer instancié à la trame d'initialisation)
- un déclencheur de lecture (booléen de la trame source)
- si le son doit être lu en boucle ou non (booléen de la trame d'initialisation)
- le gain de lecture (prise en compte des coordonnées de l'auditeur et de la source)

## 5.8 Tests avant installation

Au CRVM, il est difficile d'accéder aux machines et de faire tourner les applications quand on le souhaite, avant d'y installer le matériel audio et ViSpat, il a donc fallu simuler le cas réel au LMA. Nous avons reliés deux ordinateurs par Ethernet, celui sur lequel à été développé ViSpat et l'autre faisant office de "générateur de trames". Ainsi, des trames types auditeur et sources furent envoyées à ViSpat à une fréquence de 60 Hz afin d'apprécier le comportement du système et d'appréhender les bugs. Cela nous à permis de recalibrer certains paramètres coûteux de la carte audio (ex : I/O vector size, signal vector size). Dans le cas réel, d'autres mises au point ont été nécessaires, notamment dues au fait que le flux de données était en réalité parfois très supérieur à 60 Hz, ou bien que les trames envoyées comprenaient des guillemets (invisible pour Virtools mais apparent dans MAX/MSP).

## Chapitre 6

# Conclusion et perspectives

### 6.1 Une limite de Spat

La version de Spat utilisée pour ce travail (3.3.2 - août 2007) ne permet pas le contrôle de l'orientation de l'auditeur mais uniquement celui de la source.

Concernant l'orientation de la source, nous pouvons agir sur deux paramètres : le *yaw* (orientation de la source dans le plan horizontal) et le *pitch* (orientation de la source dans le plan vertical). N'ayant pas de moyens de contrôles similaires sur l'auditeur, si nous souhaitons simuler une modification de l'orientation de l'auditeur d'un angle  $\theta$ , la seule solution consiste donc à modifier l'orientation de la source du même angle  $\theta$ . Ceci pose un problème dans le cas suivant :

Prenons l'exemple d'un auditeur qui se situe dans un couloir, il localise une source sonore droit devant lui. S'il se tourne d'un angle  $+\frac{\pi}{4}$ , aux vues de ce qui a été dit plus haut, cela pourrait se traduire dans Spat par une rotation de la source de  $+\frac{\pi}{4}$  avec un auditeur regardant toujours droit devant lui. Le problème est qu'une réverbération sera générée à cause des murs du couloir. Cette réverbération n'aurait pas existé dans le cas où si l'auditeur avait réellement fait une rotation sur lui-même et non la source.

Ceci pose donc un problème de réalisme et nous ne pouvons pour l'instant pas le résoudre avec cette version du Spat. Cependant, il s'agit d'un cas très particulier et nous pouvons, dans un premier temps, réaliser des démonstrations sans tenir compte de cet aspect.

## 6.2 La démonstration

L'intérêt du projet est de montrer le rôle considérable que joue le son, dans un système de réalité virtuelle. Il est donc intéressant de mettre au point une démonstration qui rend compte de cette ampleur.

Spat gère plusieurs paramètres de spatialisation, il faut pouvoir les mettre en avant dans la démonstration. Il s'agit de :

- la distance (éloignement, rapprochement)
- l'azimut
- l'élévation
- la réverbération
- l'effet Doppler

Pour ce premier essai, nous avons choisi une ballade dans un château médiéval : le sujet est initialement positionné proche d'une fontaine, dans la place d'un village constitué de quelques maisonnettes, il accède à un château via un pont-levis, il peut également sortir du village pour se balader dans la nature. Nous avons sonorisé cet environnement avec six sources : la fontaine, un forgeron travaillant dans l'une des maisons, des personnes qui discutent dans une autre maison, la rivière sous le pont-levis, des animaux qui tournent autour du village au loin et un corbeau tournoyant au dessus du château. Une septième source, non spatialisée, est un enregistrement effectué dans le bois de Luminy, proche du CRVM, lu en boucle et à très bas niveau sonore afin de recréer une ambiance générale naturelle : vent, craquement de branches, oiseaux etc. Chacun des six sons spatialisés évolue en fonction de la position de l'auditeur et de la direction de son regard mettant ainsi en évidence la gestion de l'azimut et de la distance. Le corbeau qui survole le château rend compte de l'effet d'élévation. L'effet Doppler est difficile à mettre en évidence dans le cadre d'une reconstitution médiévale mais l'utilisateur peut augmenter sa vitesse de déplacement avec le joystick et entendre quel serait son environnement sonore s'il se déplaçait très rapidement. La réverbération des sources est modifiée lorsque l'on passe sous le porche du château.

Les personnes du CRVM et du CNRS ayant testé la démonstration ont été très satisfaites du résultat dans le sens où l'immersion s'accroît fortement grâce au son. Bien sûr, le design sonore (mixage des sources, travail des enregistrements, choix des échantillons sonores etc.) apparaît comme un élément important pour le rendu final. Il mérite d'être travaillé de manière à ce que les sons soient les plus réalistes et les moins monotones possibles. Mais ceci est un travail long, propre à chaque démonstration et nécessiterait, dans l'idéal, qu'il soit réalisé par un professionnel (type ingénieur du son).



### 6.3 Conclusion et perspectives

Nous avons au départ deux outils de spatialisation à notre disposition, d'une part le logiciel Virtools qui commande les scénarios visuels de la plateforme de visualisation stéréoscopique du CRVM et d'autre part le logiciel de spatialisation sonore Spat développé par l'Ircam et utilisé au LMA. Ces deux logiciels ne suivent pas les mêmes protocoles car leurs champs d'applications sont différents. Ce stage a permis d'associer ces deux systèmes au moyen d'un logiciel intermédiaire développé sur MAX/MSP : ViSpat.

Nous avons montré le rôle capital que joue le son en termes d'immersion virtuelle 3D. En effet, le son apporte une dimension supplémentaire au point qu'une scène virtuelle dotée d'un environnement sonore devient beaucoup moins réaliste une fois qu'elle en est dépourvue. Mais seule la démonstration nous a permis de juger de l'utilité du son, à l'avenir, il serait intéressant de mettre en place des procédés permettant une évaluation plus précise de notre système pour ainsi étudier les interactions : quelle influence à le son sur l'image, et inversement ?

Enfin, notre système connaît certaines limites techniques dues à l'ampleur des calculs demandés (d'où une limitation du nombre de sources à spatialiser) et à la façon dont Spat a été implémenté. Le design sonore, la gestion de la réverbération et des occlusions méritent d'être d'avantages travaillés afin d'accroître le réalisme. Certaines positions particulières génèrent encore des artefacts sonores dus au fait que nous avons eus du mal à comprendre le fonctionnement de Virtools dans sa globalité.

Cela dit, nous avons montré que cette association son/image est possible, elle est aujourd'hui en grande partie fonctionnelle et ouvre un nouveau champ d'applications pour le CRVM en vue de poursuivre son étude sur le comportement humain.

# Chapitre 7

## Annexes

### 7.1 Fiche Technique ViSpat

#### 7.1.1 Traitement des trames

##### Traitement de la trame d'initialisation

La trame d'initialisation se compose des éléments suivants :

/INIT	flag de reconnaissance (string)
int	nombre de sources sonores
int	numéro de la première source (0)
int	méthode de synthèse de la première source
si méthode 1	
string	adresse du fichier son
bool	0 : son joué en boucle, 1 : son joué une fois
si méthode 2	
..	autres paramètres
int	directivité première source
int	numéro de la deuxième source (1)
int	méthode de synthèse de la deuxième source
... ..	...

Exemple : pour 3 sources à initialiser en méthode sampling :

```
/INIT 3 0 1 "Cloche.mp3" 1 180 1 1 "Gong.WAV" 1 60 2 1 "Cymbale.wav"  
0 60
```

Ici la source 0 et la source 1 seront jouées en boucle dès l'instant où l'on appellera la fonction de lecture, la source 2 ne sera jouée qu'une seule fois à l'appel de la fonction de lecture.

La source 0 aura un angle d'ouverture de  $360^\circ$  (source omnidirectionnelle) sur le plan horizontal et vertical, les sources 1 et 2 auront un angle d'ouverture de  $120^\circ$ .

Les adresses de fichiers sont entre guillemets et peuvent comporter des espaces. Les formats acceptés sont le .mp3 et le .wav

Pour l'instant, seule la méthode de synthèse 1 à été implémentée, il s'agit de la méthode de sampling.

Le traitement de la trame d'initialisation est un processus cyclique réitéré autant de fois qu'il y'a de sources. Ci-dessous le principe du traitement de la trame d'initialisation.

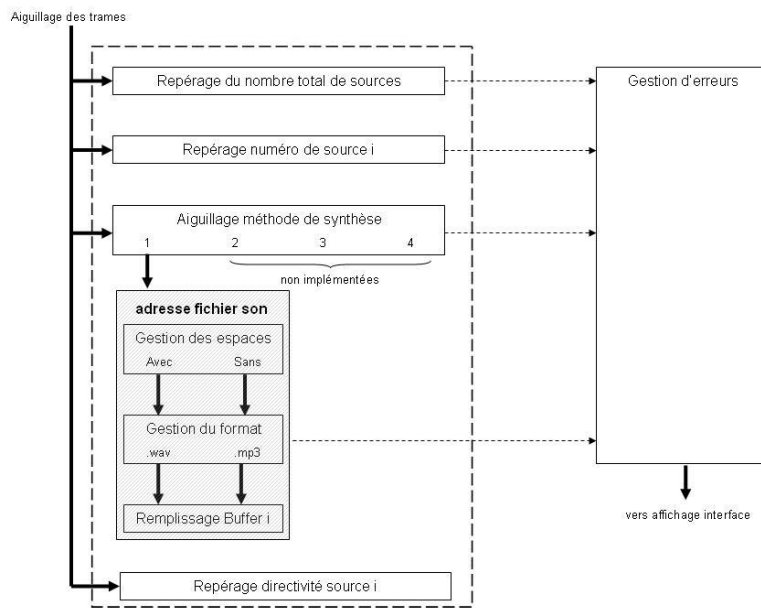


FIG. 7.1 – Schéma du traitement de la trame d'initialisation

La première étape consiste à repérer le nombre total de sources. Voir le patch *TraitementTrames* : annexe ViSpat à été implémenté pour recevoir 10 sources (9 sources spatialisées et une source d'ambiance non spatialisée). Si le nombre total de sources est autre qu'un entier compris entre 1 et 10, un message d'erreur s'affiche sur l'interface utilisateur.

Dans un deuxième temps, on repère le numéro de la première source. Cet entier doit forcément être un 0, l'entier de la source suivante un 1 etc. Si les sources sont déclarées dans le mauvais ordre, un message d'erreur apparaît.

La troisième étape consiste à aiguiller le reste de la trame selon la méthode de synthèse choisie.

La quatrième étape est celle de la gestion de l'adresse du fichier son. Elle se décompose en trois sous-étapes.

- la gestion des espaces

Il est important de savoir si l'adresse fichier contient des espaces ou non car MAX/MSP ne remplit pas ses buffers de la même manière dans les deux cas.

A chaque adresse fichier son reçue, les caractères ASCII constituant l'adresse sont converties en entiers. Chaque adresse fichier est écrite entre guillemets, mais dans MAX/MSP, si l'adresse contient des espaces, ces guillemets ne seront pas visibles une fois l'adresse converties en suite d'entiers, si l'adresse ne contient pas d'espace, ils seront visibles.

Pour chaque adresse il suffit donc de regarder son premier élément entier pour en déduire si elle contient des espaces ou non. Voir Patch *TraitementEspace* : annexe 7.8 page 56.

- la gestion du format

De même que pour les espaces, MAX/MSP ne remplit pas ses buffers de la même façon selon que le fichier son soit en .wav ou en .mp3.

Pour faire la distinction il suffit de repérer les 3 derniers caractères du fichier constituant son extension. Cependant ceux-ci peuvent contenir un guillemet selon les raisons vues plus haut. Il faut donc un traitement pour les adresses sans espace et les adresses avec espaces. Voir patch *TraitementFormat* : annexe 7.9 page 57.

- le remplissage des buffers

Les fichiers sons sont mémorisés dans des buffers. Un buffer prend comme argument l'adresse du fichier son. Voir patch *Buffer* : annexe 7.13 page 61.

En fin de boucle, il faut réitérer ce traitement pour les autres sources. C'est le rôle du patch *So2Buff-Syn1* : annexe 7.12 page 60.

### Traitement de la trame source

Une trame source se compose de la manière suivante :

/RTs	flag de reconnaissance (string)
int	numéro de la source
int	niveau sonore contextuel
float	$X_i$ abscisse de la source par rapport au CAVE (en mm)
float	$Y_i$ ordonnée de la source par rapport au CAVE (en mm)
float	$Z_i$ hauteur de la source par rapport au CAVE (en mm)
float	$R_{X_i}$ angle d'Euler selon $X_i$ par rapport au CAVE (en °)
float	$R_{Y_i}$ angle d'Euler selon $Y_i$ par rapport au CAVE (en °)
float	$R_{Z_i}$ angle d'Euler selon $Z_i$ par rapport au CAVE (en °)
bool	0 : le son n'est pas joué, 1 : le son est joué

Une nouvelle trame source est envoyée dès qu'un des paramètres change.

Le traitement d'une trame source est assez simple, il suffit de récupérer chaque entier ou flottant et de les envoyer au bon endroit de manière à interagir correctement avec le logiciel de spatialisation.

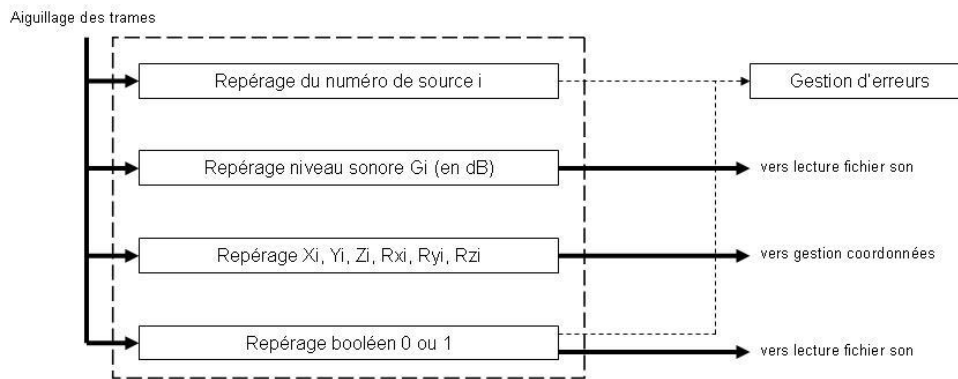


FIG. 7.2 – schéma du traitement de la trame source

Voir les patches *AiguillageRTs* : annexe 7.5 page 54 et *RecupRTs* : annexe 7.6 page 54.

### 7.1.2 Traitement de la trame auditeur

La trame auditeur se compose de la manière suivante :

/RTa	flag de reconnaissance (string)
float	$X_a$ abscisse de l'auditeur par rapport au CAVE
float	$Y_a$ ordonnée de l'auditeur par rapport au CAVE
float	$Z_a$ hauteur de l'auditeur par rapport au CAVE
float	$R_{X_a}$ angle d'Euler selon $X_a$ par rapport au CAVE
float	$R_{Y_a}$ angle d'Euler selon $Y_a$ par rapport au CAVE
float	$R_{Z_a}$ angle d'Euler selon $Z_a$ par rapport au CAVE

Sur le même modèle que la trame source, il suffit ici aussi de récupérer chaque flottant et de les envoyer au bon endroit.



FIG. 7.3 – schéma du traitement de la trame auditeur

Voir le patch RecupRTa : Annexe 7.4 à la page 53

### Traitement de la trame réverbération

Une trame réverbération se compose de la manière suivante :

/REV	flag de reconnaissance (string)
int	numéro de réverbération

A chaque entier correspond un paramétrage particulier de réverbération. Pour l'instant ViSpat accepte deux entiers, 0 : pas de réverbération, et 1 : un peu de réverbération.

La réverbération appliquée est commune à toutes les sources spatialisée.

Pour modifier la réverbération générale, on peut agir sur les points suivants (tiré du manuel de l'Ircam) :

source presence :	early sound : energy of direct sound and early room effect
source brilliance :	variation of early sound at high frequencies
source warmth :	variation of early sound at low frequencies
room presence :	late sound : energy of later reflections and reverberation
running reverberance :	early decay time
envelopment :	energy of early room effect relative to direct sound
late reverberance :	mid-frequency decay time
liveness :	relative decay time at high frequencies
intimacy :	relative decay time at low frequencies

### Traitement de la trame gain

Afin d'agir en temps réel sur le gain général de l'environnement sonore, une trame à été mise en place :

/GEN	flag de reconnaissance (string)
int	gain général (en dB)

### Traitement de la trame arrêt

La trame arrêt se compose de la manière suivante :

/ARRET	flag de reconnaissance (string)
--------	---------------------------------

Dès réception de cette trame, l'audio est coupée, les buffers sont vidés et le système est prêt à recevoir une nouvelle trame /INIT.

### 7.1.3 Gestion de l'orientation des sources dans le plan horizontal

Soient une source S et un auditeur A vus sur le plan horizontal

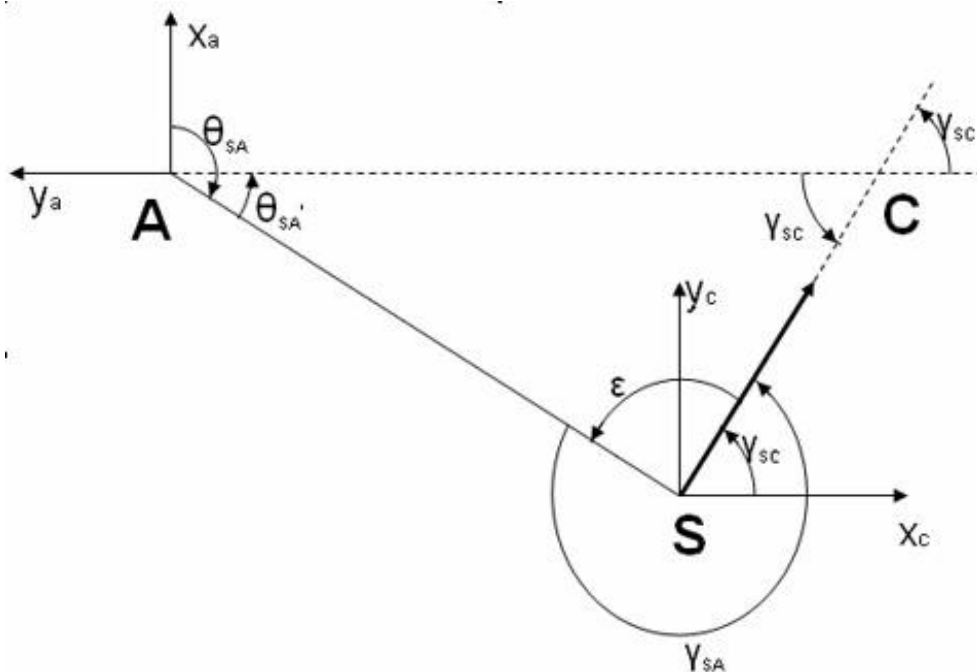


FIG. 7.4 – Principe d'asservissement de ViSpat

On décide d'attribuer à la source le repère du CAVE tel qu'une orientation de  $0^\circ$  signifiera que la source sera dirigée selon l'axe  $X_c$ . Un auditeur, quand à lui, ayant une source située à  $0^\circ$  signifie que cette source est devant lui. C'est ainsi qu'est implémenté Spat. Le repère d'orientation de l'auditeur est donc décalé de  $+90^\circ$  par rapport au repère d'orientation du CAVE.

La question est la suivante : comment faire en sorte qu'une orientation  $\gamma_{SC}$  donnée par rapport au CAVE ( n'évolue pas même si l'auditeur bouge ?

Soient les angles suivants :

- $\gamma_{SC}$  : orientation de la source par rapport au CAVE
- $\gamma_{SA}$  : orientation de la source par rapport à l'auditeur
- $\theta_{SA}$  : azimut de la source par rapport à l'auditeur

Soit C le point de croisement entre l'orientation de la source et l'axe  $Y_a$



Dans le triangle SAC on a :

$$\theta'_{SA} + \gamma_{SC} + \epsilon = 180$$

or

$$\theta'_{SA} = -\theta_{SA} - 90$$

et

$$\epsilon = 360 - \gamma_{SA}$$

Par suite,

$$\gamma_{SA} = \gamma_{SC} - \theta_{SA} + 90$$

Nous avons donc une relation permettant d'ajuster  $\gamma_{SA}$  (correspondant au paramètre Yaw dans Spat) en fonction de l'azimut de la source par rapport à l'auditeur  $\gamma_{SA}$ .

De plus, rien n'empêche de faire tourner la source sur elle-même, il suffit de faire varier  $\gamma_{SC}$ , correspondant à  $R_Z$  dans la trame /RTs

#### 7.1.4 Gestion de l'azimut et de l'élévation d'une source

Soit une source localisée dans le repère de l'auditeur à la position  $X_{s0}$ ,  $Y_{s0}$ ,  $Z_{s0}$ .

La distance entre l'auditeur et la source est donnée par la relation :

$$\rho = \sqrt{X_{s0}^2 + Y_{s0}^2 + Z_{s0}^2} \quad (7.1)$$

L'azimut de la source par rapport à l'auditeur est donnée par :

$$\phi = \text{Arctan}\left(\frac{Y_{s0}}{X_{s0}}\right) \quad (7.2)$$

L'élévation de la source par rapport à l'auditeur est donnée par :

$$\theta = \frac{\pi}{2} - \text{Arccos}\left(\frac{Z_{s0}}{\rho}\right) \quad (7.3)$$

Soient  $R_x, R_y, R_z$  les trois rotations appliquées au repère de l'auditeur respectivement sur les axes  $x, y, z$ .

Le but de la démarche est d'exprimer le nouvel azimut et la nouvelle élévation que ces trois rotations engendrent. Figure 7.5 page 49.

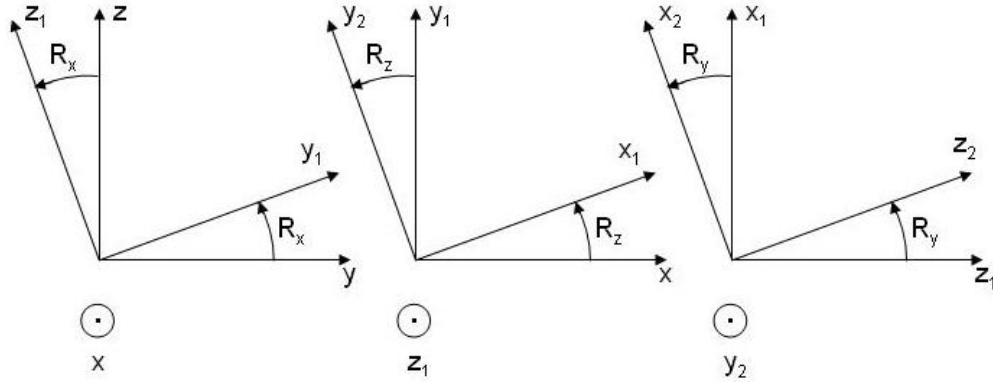


FIG. 7.5 – Schéma d'une rotation sur X puis sur Z puis sur Y

**Étape 1** En premier lieu, on exprime le nouveau repère en fonction de l'ancien.

En considérant les propriétés de Virtools, nous pouvons découper toute rotation quelconque en composée de trois rotations sur X puis sur Z puis sur Y comme le montre la figure 7.5, aux vues de ces trois schémas nous obtenons les six relations suivantes :

$$\begin{aligned}
 \vec{y}_1 &= \cos R_x \cdot \vec{y}_0 + \sin R_x \cdot \vec{z}_0 \\
 \vec{z}_1 &= \cos R_x \cdot \vec{z}_0 - \sin R_x \cdot \vec{y}_0 \\
 \vec{x}_1 &= \cos R_z \cdot \vec{x}_0 + \sin R_z \cdot \vec{y}_1 \\
 \vec{y}_2 &= \cos R_z \cdot \vec{y}_1 - \sin R_z \cdot \vec{x}_1 \\
 \vec{z}_2 &= \cos R_y \cdot \vec{z}_1 + \sin R_y \cdot \vec{x}_1 \\
 \vec{x}_2 &= \cos R_y \cdot \vec{x}_1 - \sin R_y \cdot \vec{z}_1
 \end{aligned}$$

En développant ces relations, on obtient, sous forme de matrice :

$$\begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \cdot \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} \quad (7.4)$$

où

$$\begin{aligned}
a &= \cos R_z \cdot \cos R_y \\
b &= \cos R_y \cdot \sin R_z \cdot \cos R_x + \sin R_y \cdot \sin R_x \\
c &= \cos R_y \cdot \sin R_z \cdot \sin R_x - \sin R_y \cdot \cos R_x \\
d &= -\sin R_z \\
e &= \cos R_z \cdot \cos R_x \\
f &= \sin R_x \cdot \cos R_z \\
g &= \sin R_y \cdot \cos R_z \\
h &= \sin R_y \cdot \sin R_z \cdot \cos R_x - \cos R_y \cdot \sin R_x \\
i &= \sin R_y \cdot \sin R_z \cdot \sin R_x + \cos R_y \cdot \cos R_x
\end{aligned}$$

**Étape 2** Il faut maintenant pouvoir, à l'inverse, exprimer l'ancien repère  $(x_0, y_0, z_0)$  en fonction du nouveau  $(x_2, y_2, z_2)$ .

Cela revient à inverser la matrice de passage de la relation 7.4 et obtenir

$$\begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} = \frac{1}{\det} \cdot \begin{pmatrix} j & k & l \\ m & n & o \\ p & q & r \end{pmatrix} \cdot \begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} \quad (7.5)$$

où

$$\begin{aligned}
j &= ei - fh \\
k &= ch - bi \\
l &= bf - ce \\
m &= fg - di \\
n &= ai - cg \\
o &= cd - af \\
p &= dh - eg \\
q &= bg - ah \\
r &= ae - bd
\end{aligned}$$

et

$$\det = a(ei - fh) - b(di - fg) + c(dh - eg)$$

**Étape 3** Une fois que l'on sait exprimer l'ancien repère en fonction du nouveau, on recalcule la position de la source dans ce nouveau repère.

Ce qui donne :

$$X_{s2} = \frac{1}{\det}(X_{s0}.j + Y_{s0}.m + Z_{s0}.p)$$

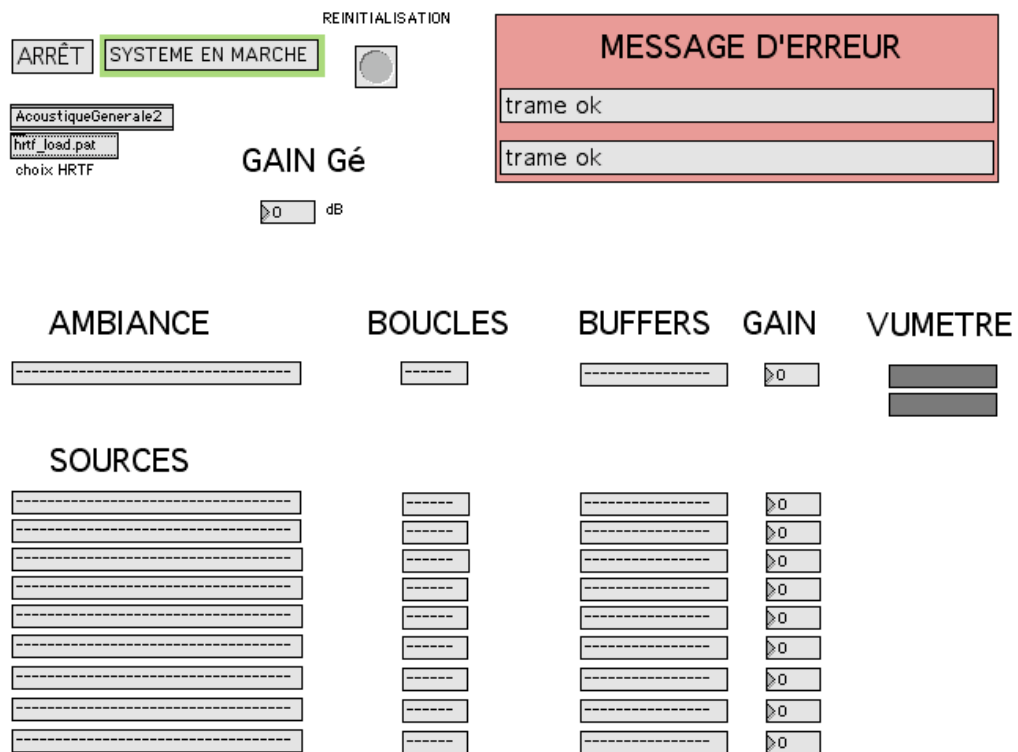
$$Y_{s2} = \frac{1}{\det}(X_{s0}.k + Y_{s0}.n + Z_{s0}.q)$$

$$Z_{s2} = \frac{1}{\det}(X_{s0}.l + Y_{s0}.o + Z_{s0}.r)$$

**Étape 4** La dernière étape consiste à recalculer le nouvel azimut et la nouvelle élévation sous le modèle des relations 7.2 et 7.3.

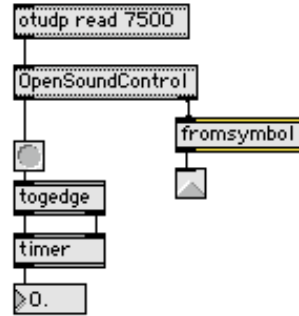
## 7.2 patch Main

Interface du logiciel réalisé. Pour chaque son : affichage de son nom, de son mode de lecture (simple ou en boucle), vérification de l'état du buffer dans lequel il est stocké (prêt ou non), son gain (en dB). Un cadran d'erreur permet de repérer le type de l'erreur et sa cause. Un vumètre permet de visualiser le niveau général de la scène sonore, ajustable par *Gain Gé*. Un bouton d'arrêt permet de réinitialiser tous les buffers et de stopper les flux audio. Un bouton *hrtfloadpat* permet de choisir ses HRTF. Un bouton *AcoustiqueGenerale* permet de régler la réverbération de l'environnement simulé (voir 7.21 page 69).



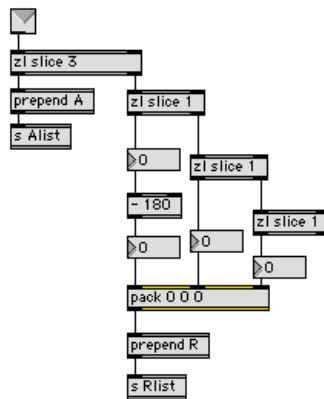
### 7.3 patch RecepteurUDP

Permet de récupérer les trames UDP provenant du système informatique du CRVM.



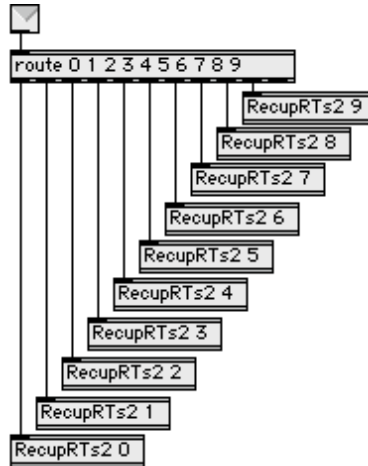
### 7.4 patch RecupRTa

Isole les 6 paramètres de la trame auditeur et les communique au patch *ConvCooPos* (7.14 page 62).



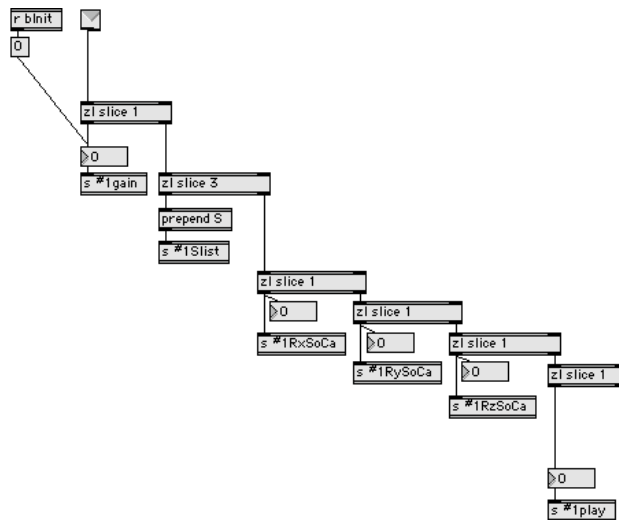
## 7.5 patch AiguillageRTs

Dissocie les trames sources selon leur numéro, chaque trame source est stockée dans une variable différente pour la suite du traitement.



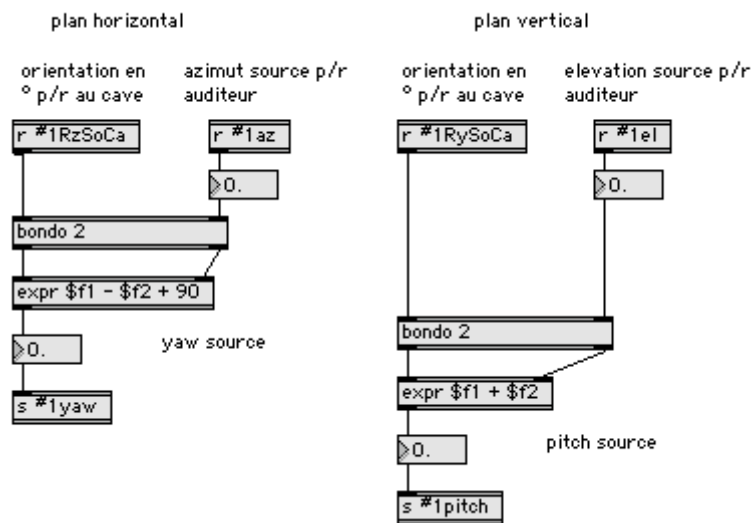
## 7.6 patch RecupRTs

Récupération des 8 paramètres de la trame source. On communique le gain et le mode de lecture (simple ou en boucle) au patch *LecS* (7.20 page 68), les 3 positions cartésiennes au patch *ConvCooPos* (7.14 page 62), les 3 orientations par rapport au CAVE au patch *YawPitch* (7.7 page 55) et au patch *CooControl* (7.18 page 66).



## 7.7 patch YawPitch

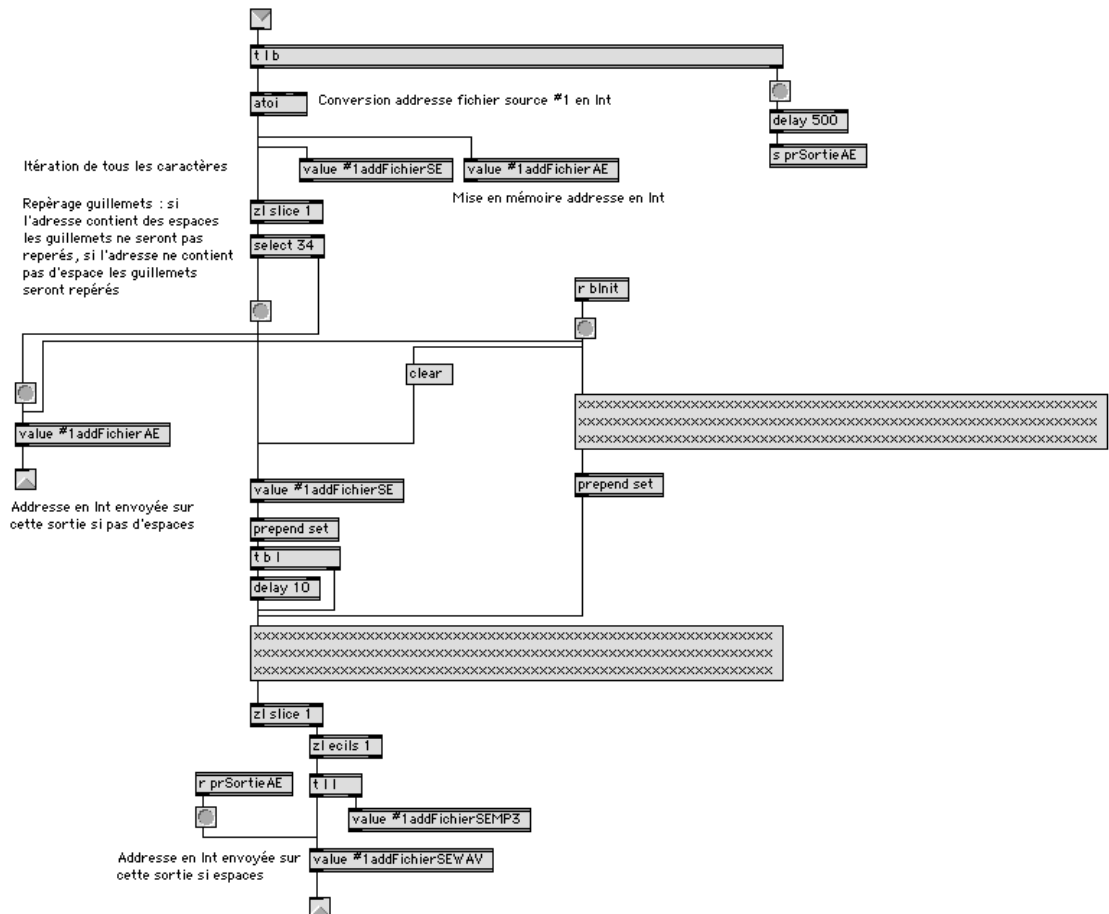
Par défaut l'orientation d'une source est invariante dans le repère de l'auditeur, ce patch permet de fixer son orientation dans le plan horizontal (yaw) et vertical (pitch) par rapport au CAVE et non plus par rapport à l'auditeur (voir section *Une Limite de Spat* 6.1 page 38).





## 7.8 patch TraitementEspace

Dans MAX/MSP, la commande de stockage d'un son n'est pas la même selon que chemin d'accès au fichier son contienne des espaces ou non. Ce patch exécute la bonne commande en fonction de l'écriture du chemin d'accès.



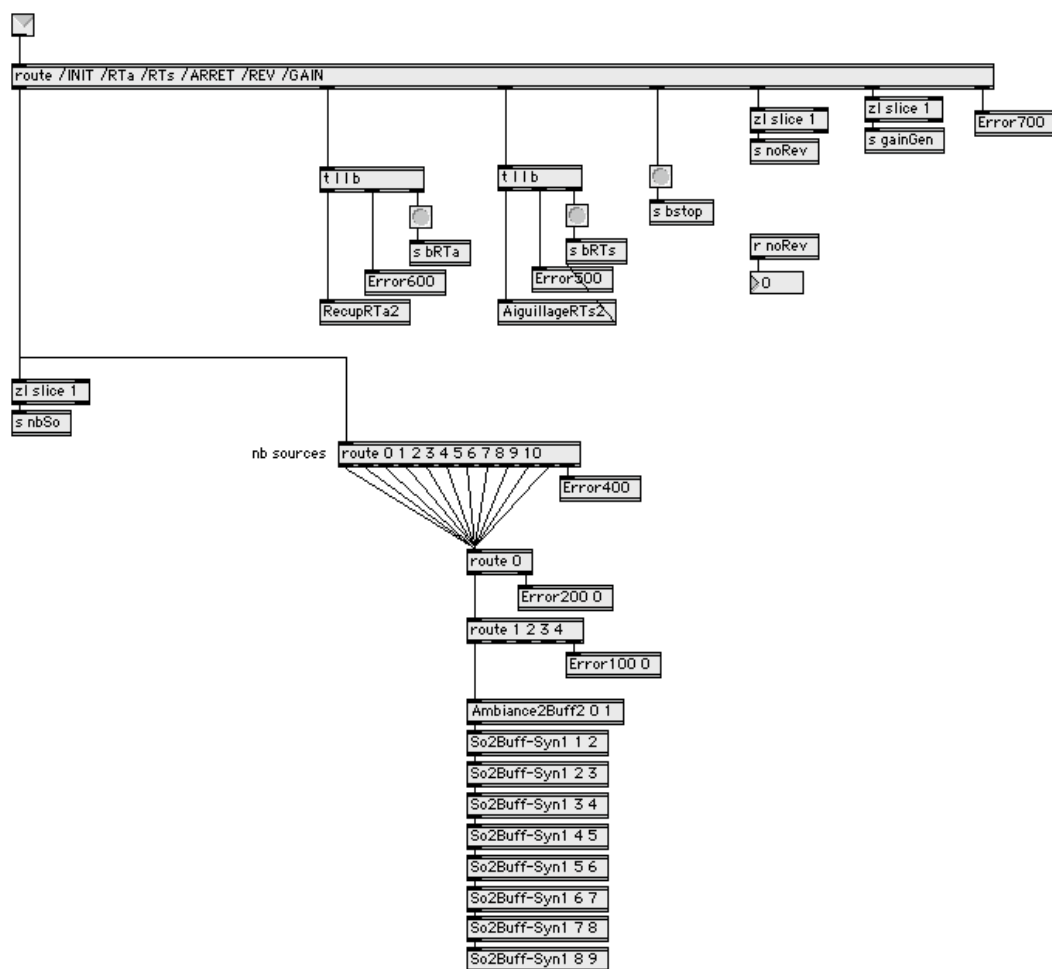
## 7.9 patch TraitementFormat

Dans MAX/MSP, la commande de stockage d'un son n'est pas la même selon que le son soit en .wav ou en .mp3. Ce patch repère l'extension du fichier son et exécute la bonne commande en fonction du format.



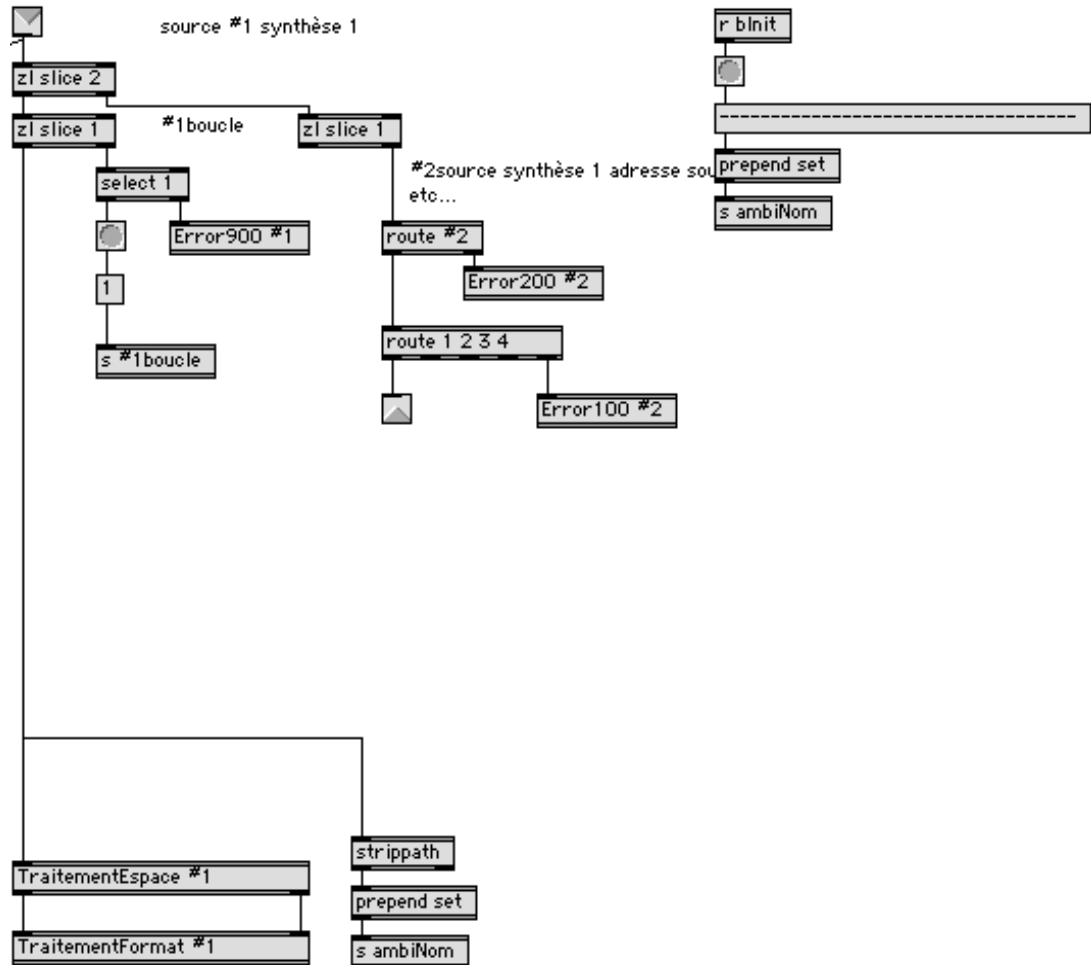
## 7.10 patch TraitementTrames

Les trames reçues par le patch *RecupUDP* doivent être identifiées et redirigées vers le traitement qui convient en fonction de leur flag. C'est le rôle de ce patch : la trame d'initialisation est dirigée vers les patches *Ambiance2Buff* (7.11 page 59) et *So2Buff-Syn1* (7.12 page 60), la trame auditeur est dirigée vers le patch *RecupRTa* (7.4 page 53), la trame sources sont dirigées vers le patch *AiguillageRTs* (7.5 page 54), la trame arrêt envoie une commande d'interruption du flux audio et réinitialise tous les buffers, les trames reverb et gain ne contiennent qu'un paramètre chacun, ils sont stockés dans des variables, enfin si aucune des trames ne correspond aux modèles précédents, un message d'erreur est généré par le patch *error700* (7.28 page 73).



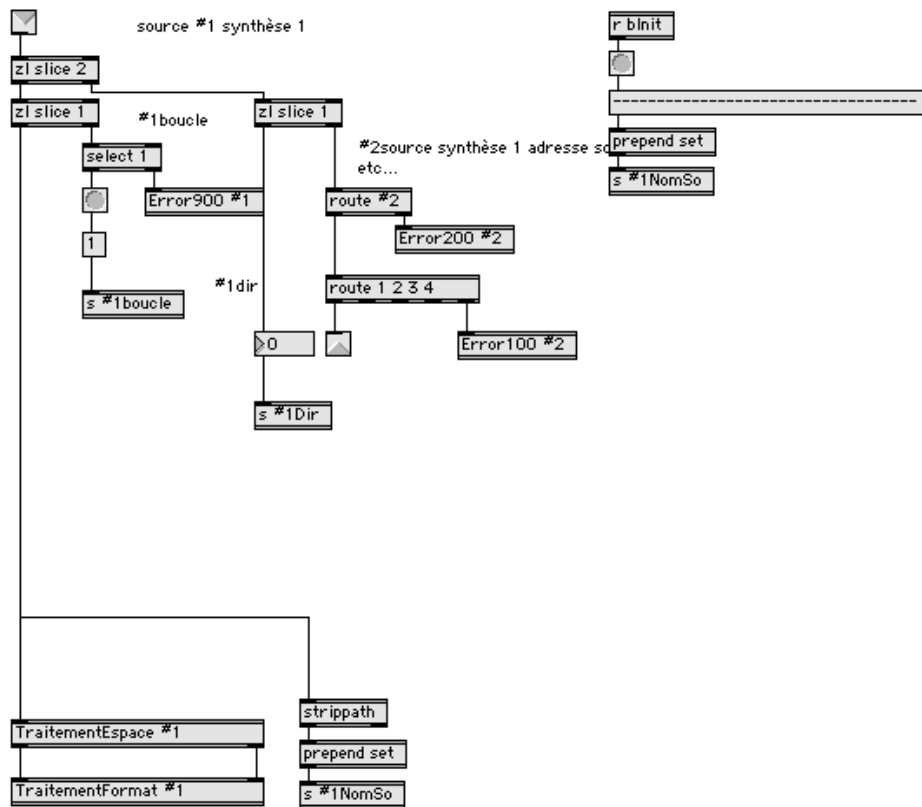
## 7.11 patch `Ambiance2Buff`

Traitement de la chaîne de caractère correspondant au chemin d'accès du fichier son contenue dans la trame source 0 (son d'ambiance). Gestion des espaces et du format pour que le fichier soit stocké dans un buffer, affichage du nom de fichier dans l'interface *Main*.



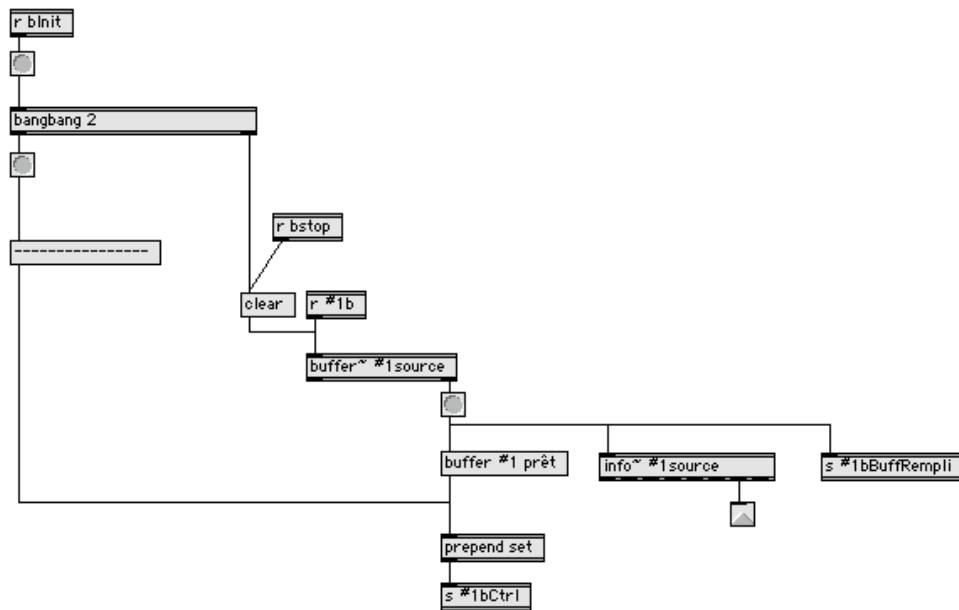
## 7.12 patch So2Buff-Syn1

Traitement de la chaîne de caractère correspondant au chemin d'accès du fichier son contenue dans les trames sources. Gestion des espaces et du format pour que les fichiers soient stockés dans des buffers, affichage des noms de fichiers dans l'interface *Main*, isolement du paramètre "directivité" pour l'envoyer au patch *SoSpat* (7.16 page 65).



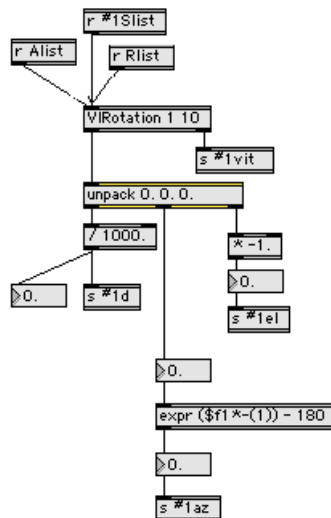
## 7.13 patch Buffer

Stockage de chaque son dans un buffer. Chaque buffer est vidé après initialisation. Envoi d'une information sur la longueur de la source sonore (en ms) au patch *LecS* pour la lecture en boucle (7.20 page 68). Affichage "buffer i prêt" sur l'interface lorsque le son i est correctement chargé et peut être lu (patch *Main* 7.2 page 52).



## 7.14 patch ConvCooPos

Récupération des paramètres nécessaires aux changements de repères et conversions (positions, orientations des sources et de l'auditeur) pour les envoyer dans le patch *ViRotation* qui gère l'ensemble de ces calculs (voir suivant). Envoi des résultats au patch *SoSpat* (7.16 page 65).



## 7.15 code Virotation

Patch codé en C pour améliorer les temps de calculs. Changement de repère sur les sources afin de les obtenir dans le repère de l'auditeur et non plus du CAVE. Conversion des coordonnées cartésiennes en coordonnées sphériques. Propagation des nouvelles positions obtenues pour des variations supérieures à un seuil fixé en paramètre d'entrée.

```
void VIR_calc(t_VIR *x)
{ float theta;
float Xm,Ym,Zm;
float Dist,Azi,Elev;
float Rx,Ry,Rz;
float cx, cy, cz, sx, sy, sz;
float a,b,c,d,e,f,g,h,i;
float det,eifh,dheg,fgdi;
float a1,b1,c1,d1,e1,f1,g1,h1,i1;
float Xm2,Ym2,Zm2;
```

```

float Dist2,Azi2,Elev2;
float deltaDist,deltaAzi,deltaElev;
t_atom myList[3];
// Conversion coordonnées Source/CAVE Auditeur/CAVE -> Source/Auditeur
Xm = x->Xm - x->Xa;
Ym = x->Ym - x->Ya;
Zm = x->Zm - x->Za;
// Passage cartésien -> sphérique
Dist = sqrt(Xm*Xm + Ym*Ym + Zm*Zm); // distance

if ((Ym==0.) && (Xm>0.)) Azi = PIs2;

else if ((Ym==0.) && (Xm<=0)) Azi = MPiS2;
Azi = -atan2(Xm,Ym); // Azimut en radians
Azi = Azi * RADIANT2DEGREE; // conversion en degrés
// Elevation
if (Dist == 0.) theta = PIs2;
else theta = acos(Zm/Dist);
Elev = PIs2 - theta;
Elev = Elev * RADIANT2DEGREE; // conversion en degrés
// Conversion des angles de rotation : degrés -> radians
Rx = x->Rx * DEGREE2RADIANT;
Ry = x->Ry * DEGREE2RADIANT;
Rz = x->Rz * DEGREE2RADIANT;
cx = cos(Rx);
sx = sin(Rx);
cy = cos(Ry);
sy = sin(Ry);
cz = cos(Rz);
sz = sin(Rz);

// matrice passage : nouveau repère exprimé en fonction de l'ancien
a = cy*cz;
b = cx*cy*sz + sy*sx;
c = cy*sz*sx - sy*cx;
d = -sz;
e = cz*cx;
f = cz*sx;
g = sy*cz;
h = sy*sz*cx - cy*sx;
i = sy*sz*sx + cy*cx;
// determinant
eifh = e*i-f*h;
fgdi = f*g - d*i;

```



```

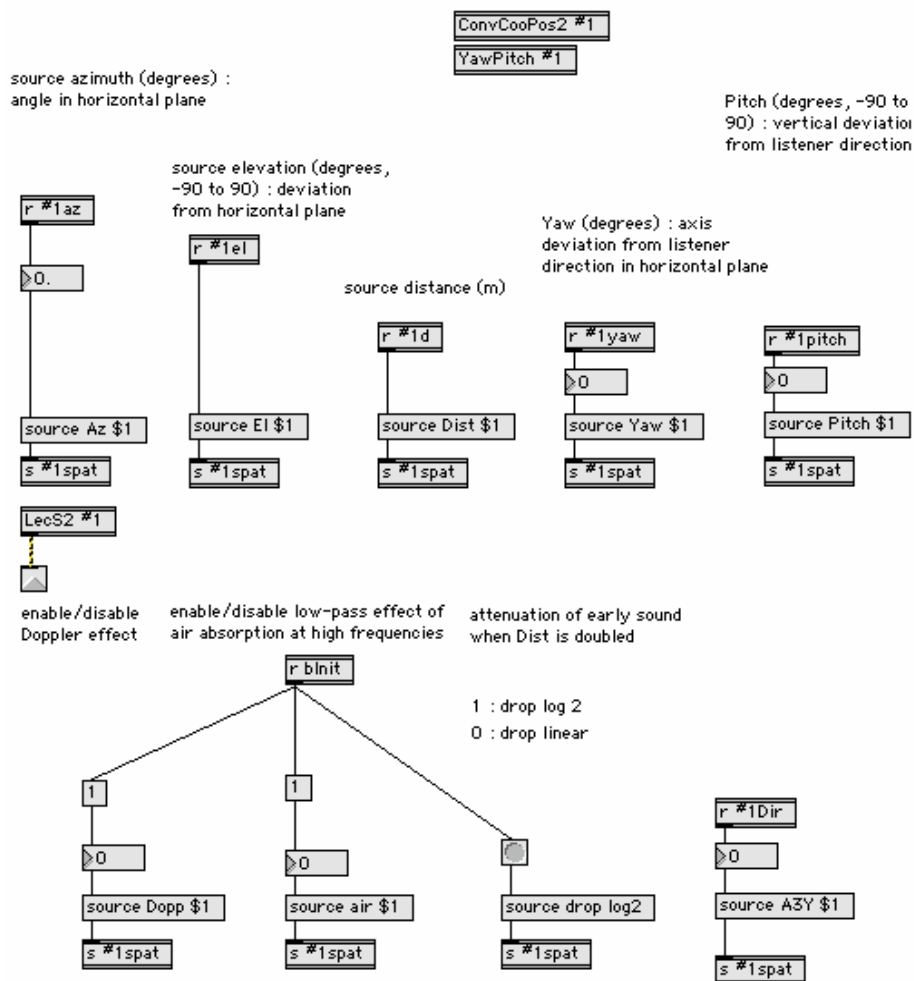
dheg = d*h-e*g;
det = a*eifh + b*fgdi + c*dheg;
// Matrice de passage ancien repère en fonction du nouveau
a1 = (eifh)/det;
b1 = (c*h - b*i)/det;
c1 = (b*f - c*e)/det;
d1 = (fgdi)/det;
e1 = (a*i - c*g)/det;
f1 = (c*d - a*f)/det;
g1 = (dheg)/det;
h1 = (b*g - a*h)/det;
i1 = (a*e - b*d)/det;
// Coordonnées de la source dans le nouveau repère
Xm2 = Xm*a1+Ym*d1+Zm*g1;
Ym2 = Xm*b1+Ym*e1+Zm*h1;
Zm2 = Xm*c1+Ym*f1+Zm*i1;
// Passage cartésien -> sphérique
Dist2 = sqrt(Xm2*Xm2 + Ym2*Ym2 + Zm2*Zm2);
if ((Ym2==0.) && (Xm2>0.)) Azi2 = PIs2;
else if ((Ym2==0.) && (Xm2<=0)) Azi2 = MPis2;
Azi2 = -atan2(Xm2,Ym2); // Azimut en radians
Azi2 = Azi2 * RADIAN2DEGREE;
if (Dist2 == 0.) theta = PIs2;
else theta = acos(Zm2/Dist2);
Elev2 = PIs2 - theta;
Elev2 = Elev2 * RADIAN2DEGREE; // conversion en degrés

// Si les variations sont assez grandes, on transmet
deltaDist = abs(Dist2 - x->Dist);
deltaAzi = abs(Azi2 - x->Azi);
deltaElev =abs(Elev2 - x->Elev);
if ((deltaDist>x->varDist)|| (deltaAzi>x->varAngle)|| (deltaElev>x->varAngle)) {
SETFLOAT(myList,Dist2);
SETFLOAT(myList+1,Azi2);
SETFLOAT(myList+2,Elev2);
outlet_list(x->outlet1_ptr,0L,3,&myList) ;
x->Dist = Dist2;
x->Azi = Azi2;
x->Elev = Elev2;
}
}

```

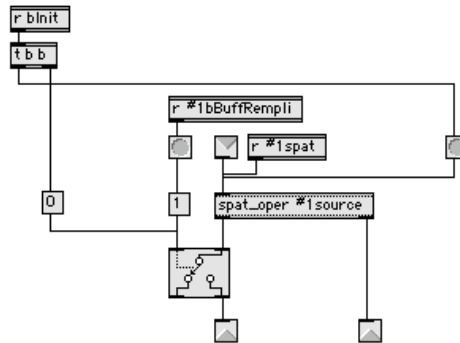
## 7.16 patch SoSpat

Récupération des paramètres de la source et envoi de ces paramètres à *SpatControl* (7.17 page 66). Ces paramètres sont propres à chaque source et sont repérés par rapport à l'auditeur. Ainsi pour chaque source on envoie son azimuth, son élévation, sa distance, son orientation (verticale et horizontale), sa directivité, l'application ou non de l'effet doppler, la prise en compte ou non de l'absorption de l'air, l'atténuation des échos précoces avec la distance (linéaire ou logarithmique).



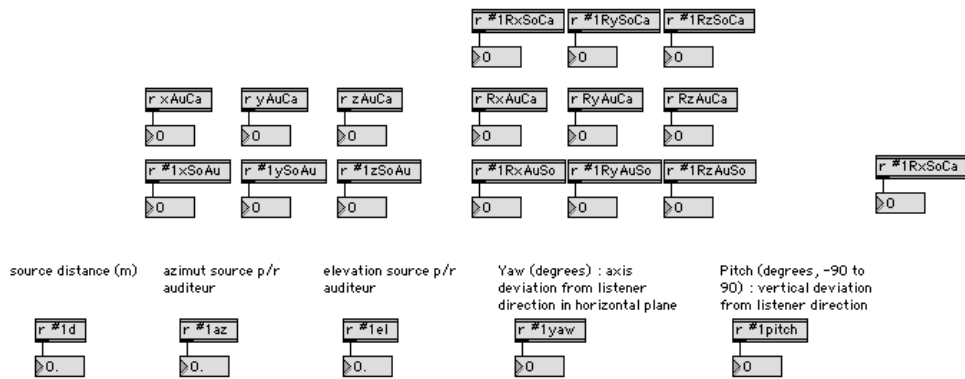
## 7.17 patch SpatControl

Récupération des paramètres cités précédemment et les injecte dans Spat dès lors que l'on est assuré que le buffer correspondant au son à filtrer ne soit pas vide.



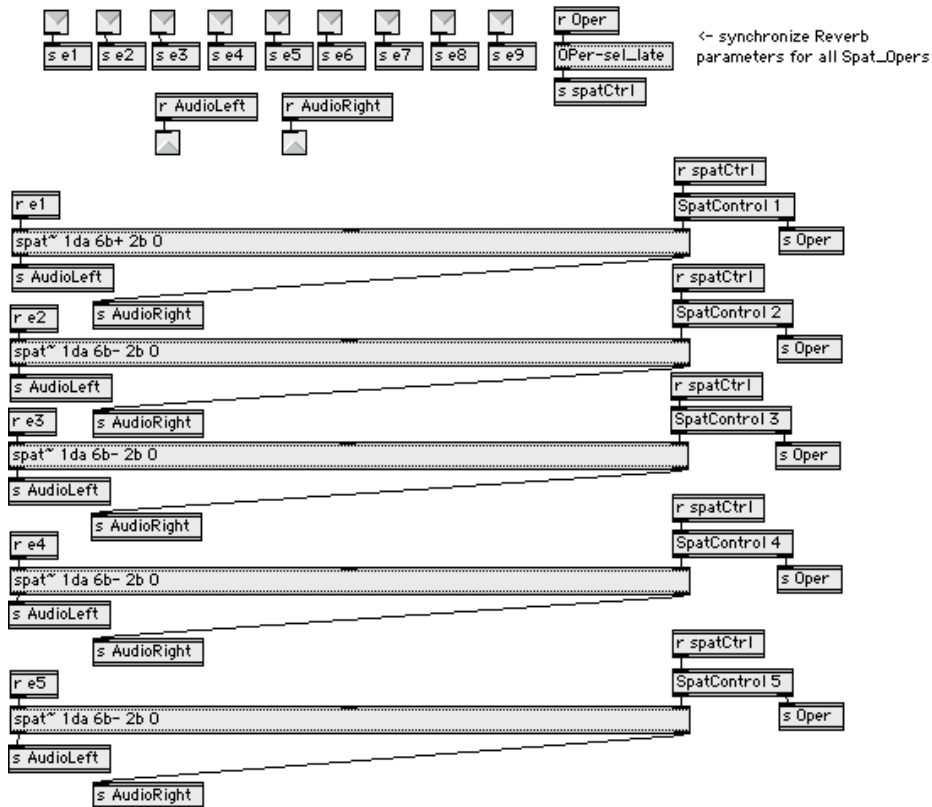
## 7.18 patch CooControl

Interface intermédiaire permettant de suivre l'évolution de chaque coordonnée de l'auditeur et des sources avant et après conversion.



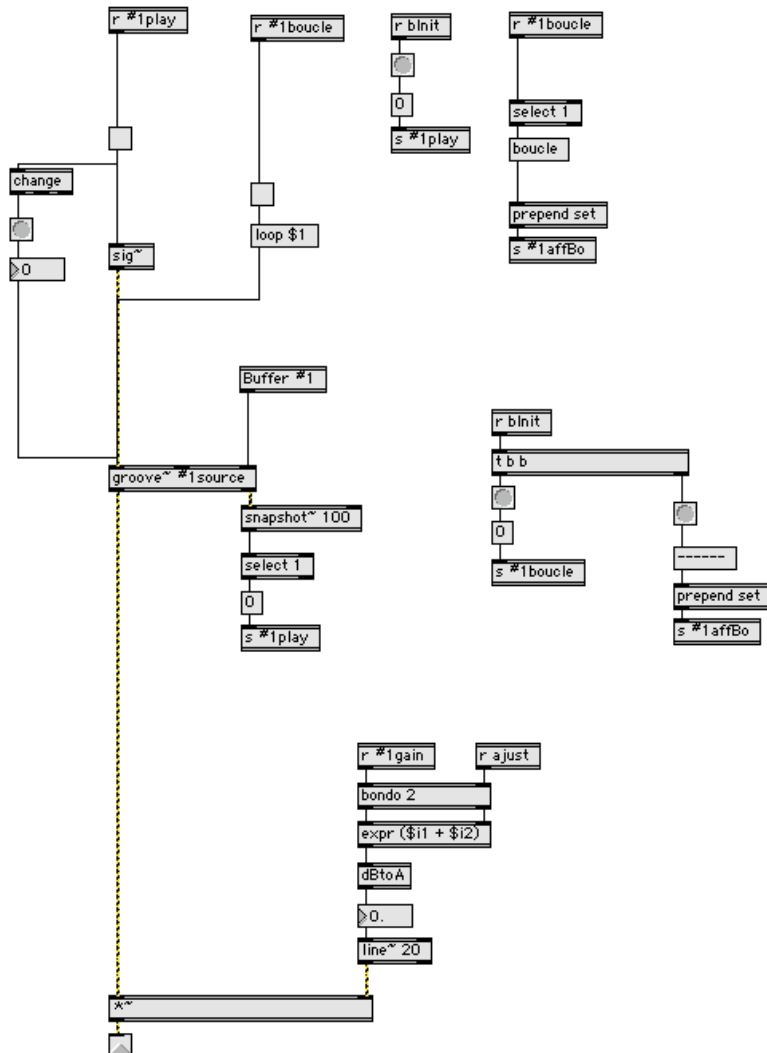
## 7.19 patch SpatReverb

Factorisation de la réverbération afin d'optimiser les temps de calcul : la réverbération est calculée une seule fois et est appliquée à tous les sons au lieu d'être calculée indépendamment pour chaque son.



## 7.20 patch LecS

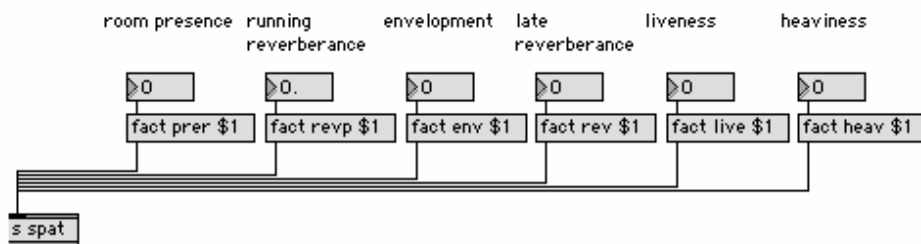
Génère le flux audio en gérant : la lecture ou l'arrêt, la lecture en boucle, le gain. Travaille sur le son stocké dans son buffer (voir le patch *Buffer* 7.13 page 61). Affiche sur l'interface si le son est joué en boucle.



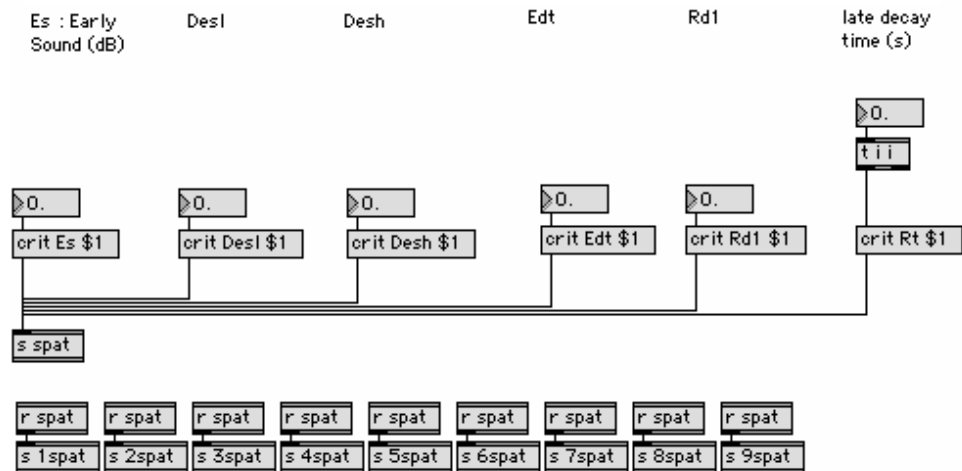
## 7.21 patch AcoustiqueGénérale

Permet d'agir sur un grand nombre de paramètres de réverbération appliqués aux sons spatialisés. Ce sont ces paramètres que la trame de réverbération devra contrôler. Pour l'instant la gestion reste manuelle.

### PERCEPTUAL FACTORS FOR ALL SOURCES

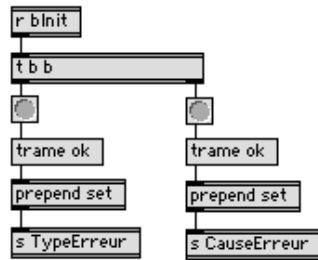


### OBJECTIVE CRITERIA FOR ALL SOURCES



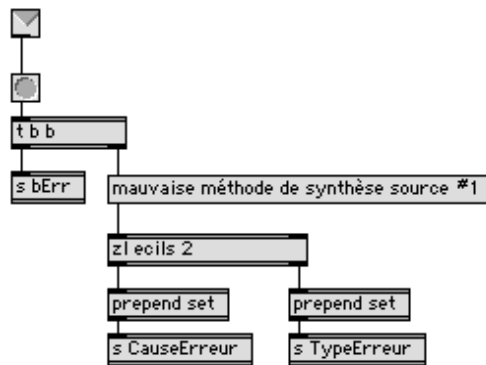
## 7.22 patch Error000

Initialise le message d'erreur à *trame ok* dans l'interface lorsqu'une nouvelle expérience commence.



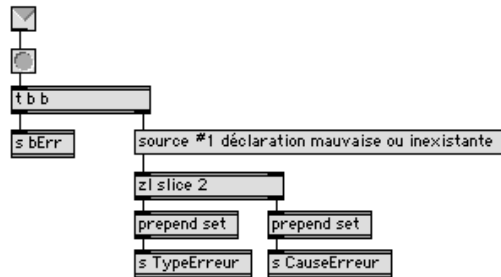
## 7.23 patch Error100

Affiche un message d'erreur dans l'interface si la méthode de synthèse de la trame source n'est pas reconnue. Déclaration du numéro de la source correspondant à la trame défaillante.



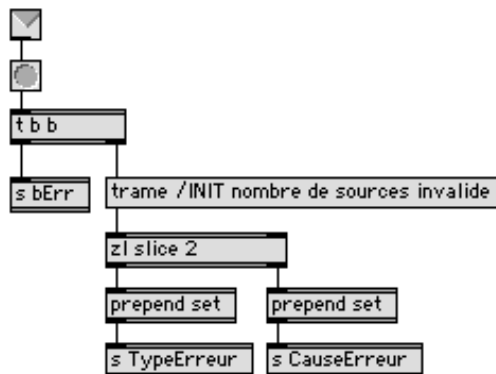
## 7.24 patch Error200

Affiche un message d'erreur dans l'interface si le chemin d'accès du fichier son n'est pas reconnu. Déclaration du numéro de la source correspondant à la trame défaillante.



## 7.25 patch Error400

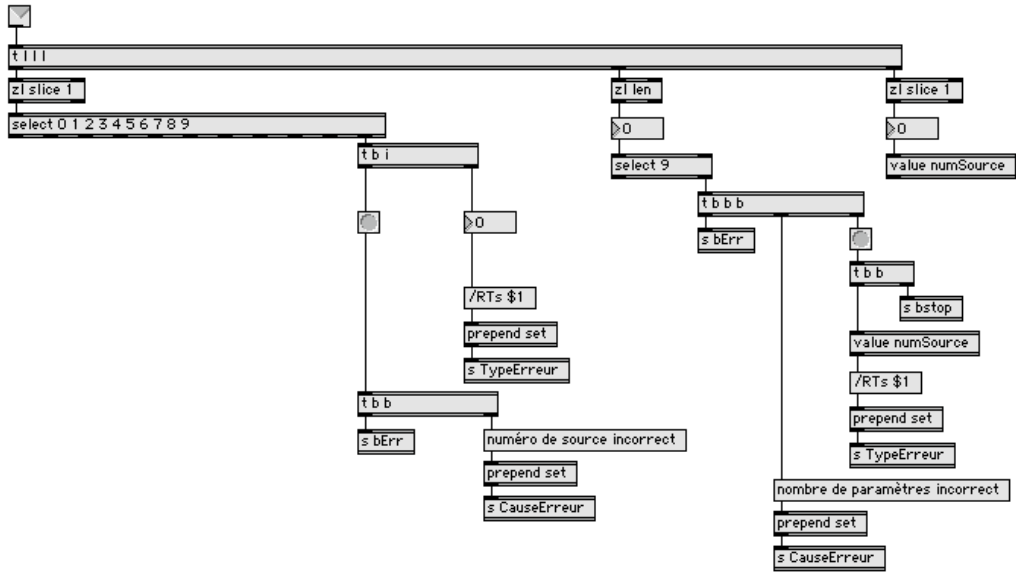
Affiche un message d'erreur lorsque le nombre de sources déclarés dans la trame d'initialisation diffère du nombre de sources réelles.





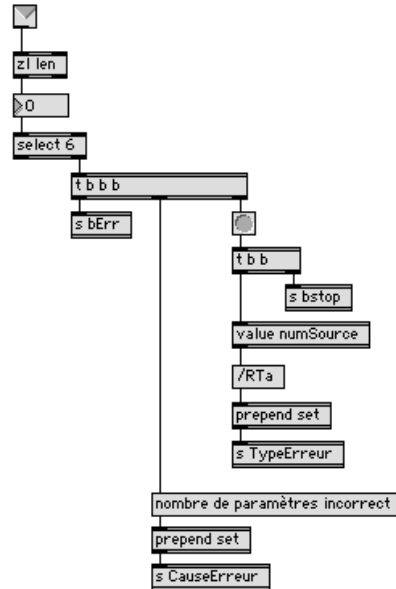
## 7.26 patch Error500

Affiche un message d'erreur si le numéro d'une source est incorrect (négatif, nul ou supérieur à 9). Affiche un message d'erreur si le nombre de paramètres d'une trame source est incorrect. Déclaration du numéro de la source correspondant à la trame défaillante.



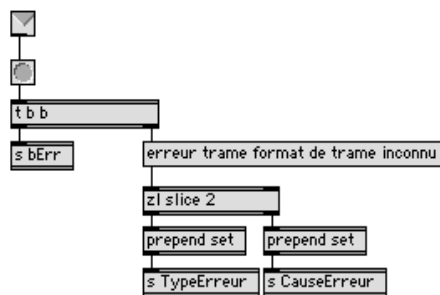
## 7.27 patch Error600

Affiche un message d'erreur si le nombre de paramètres de la trame auditeur est incorrect.



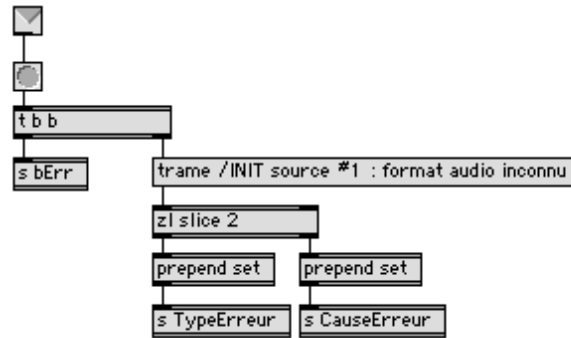
## 7.28 patch Error700

Affiche un message d'erreur si le format de la trame est inconnu (mauvais flag).



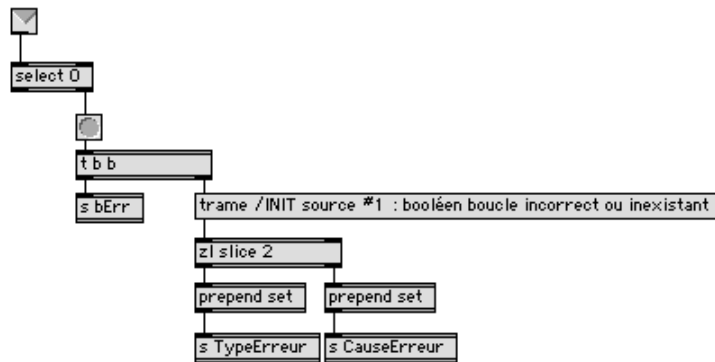
## 7.29 patch Error800

Affiche un message d'erreur si le format de la source n'est ni du mp3 ni du wav. Déclaration du numéro de la source correspondant à la trame défaillante.



## 7.30 patch Error900

Affiche un message d'erreur si le booléen de lecture en boucle d'une des sources est incorrect. Déclaration du numéro de la source correspondante.



## 7.31 Fiche d'utilisation ViSpat

Cette fiche à été remise à l'équipe du CRVM. Elle récapitule les principaux points à connaître pour maîtriser le système audio (logiciel + matériel prêté par le CNRS).

### ViSpat

Au démarrage du Mac, ViSpat apparaît automatiquement à l'écran et est prêt à recevoir les instructions du maître.

Gain Gé : ajuste le volume général de la scène.

Message d'erreur : 2 lignes, la première pour le type d'erreur, la deuxième pour la cause.

Vumètre : informe sur le niveau sonore, ne jamais taper dans le rouge.

Bouton arrêt : éteint l'audio et réinitialise le système (affichage + vide buffers).

Choix HRTF : permet de choisir une HRTF parmi la banque proposée.

Acoustique Générale : permet de faire varier les paramètres acoustiques de la scène.

### Casque

Emetteur sur la sortie jack du Mac (devant). Casque sur la sortie jack du récepteur. Ne pas placer l'antenne trop proche du mur en béton.

Allumer l'émetteur : bouton ON Allumer le récepteur : tourner bouton vol.

Eteindre l'émetteur : 2 sec sur ON Eteindre le récepteur : bouton vol sur 0.

Mettre le casque dans le bon sens.

## Chapitre 8

# Bibliographie

- [1] Robert Hamilton, Maps and Legends : FPS-Based Interfaces for Composition and Immersive Performance, Center for Computer Research in Music and Acoustics (CCRMA), Department of Music, Stanford University, 2007.
- [2] Henrik Moller, Fundamentals of Binaural Technology, Acoustics Laboratory, Aalborg University, Aalborg, Denmark, 1992.
- [3] Veronique Larcher et Jean-Marc Jot, Techniques d'interpolation de filtres audio-numériques : Application à la reproduction spatiale des sons sur écouteurs, Congrès Français d'Acoustique, Marseille, France, Avril 1997.
- [4] Véronique Larcher, Thèse : Techniques de spatialisation des sons pour la réalité virtuelle, mai 2001.
- [5] Daniel Courville, Procédé et systèmes d'enregistrement et de reproduction sonore en trois dimensions, 1993.
- [6] Bénédicte Allier, Rapport de Stage : Le Cosmophone : Illustration sonore d'événements cosmiques, juin 1999.
- [7] Sylvain Busson, Thèse : Individualisation d'indices acoustiques pour la synthèse binaurale, janvier 2006.
- [8] Georges Canevet, Perception Auditive Spatiale, Données psychoacoustiques et modélisation, Laboratoire de Mécanique et d'Acoustique, CNRS Marseille, décembre 1987.
- [9] Jean-Marc Jot, Thèse : Etudes et réalisation d'un spatialisateur de sons par modèles physique et perceptifs, Ecole Nationale Supérieure des Télécommunications, septembre 1992.
- [10] G.F. Kuhn, Model for the interaural time differences in the azimuthal plane. J. Acoust. Soc. Am., Juillet 1977

## Chapitre 9

### Références Sites

1. <http://w3lma.cnrs-mrs.fr>
2. <http://www.realite-virtuelle.univmed.fr>
3. <http://www.wikipedia.org>
4. <http://opensoundcontrol.org>
5. <http://www.cycling74.com>
6. <http://forumnet.ircam.fr/692.html>