Gabriel Malgouyard

Analyse et synthèse de sons environnementaux

Mémoire de stage de recherche ATIAM – filière SAR, 2011

Encadrants :
Amaury La Burthe - Audiogaming
Emmanuel Saint-James - UPMC

AudioGaming
NATURAL BORN INTERACTIVE

Résumé

Un nouvel outil de sonorisation d'environnements virtuels en temps réel est présenté : contrairement à l'approche traditionnelle de rejeu d'échantillons, il s'agit ici de synthétiser à la volée des sons environnementaux.
À l'aide d'une banque de sons enregistrés au début de ces travaux, une analyse perceptive comportant une expérience de classification par similarité et de verbalisation libre est mise en œuvre. Cette analyse permet de connaître les attributs perceptifs les plus courants pour ce type de son et de les relier à des descripteurs du signal acoustique.
Un modèle de synthèse de sons environnementaux est créé à partir des données analysées. Afin de respecter les contraintes de performance liées au rendu temps réel, ce modèle est implémenté en langage C, qui possède le double avantage d'être compilé et répandu sur toutes les plates-formes ciblées.
Par rapport au prototype précédent le modèle développé offre un rendu perceptif équivalent, des contrôles plus intuitifs et réclame dix fois moins de puissance de calcul sur le PC cible. Il est également portable sur la plupart des PC et consoles de salon existants.

Table des matières

Introduction

Les environnements virtuels font aujourd'hui partie de notre quotidien : ils sont présents dans un très large spectre d'application s'étendant du jeu vidéo aux simulateurs de formation professionnelle. Si tous ces environnements ne visent pas un réalisme parfait, tous en revanche tentent de faire naître le meilleur sentiment d'immersion possible.
L'interactivité constitue l'un des moyens d'améliorer ce sentiment d'immersion. Elle est obtenue par un retour visuel des évènements de l'utilisateur, mais également par le rendu audio des sons *ad hoc* – des bruits d'ambiance comme le vent, le murmure d'une foule, et des bruits d'interaction dus à l'utilisateur comme les bruits de pas, les chocs d'objets... La technique traditionnellement utilisée consiste à rejouer sur chaque évènement des échantillons préenregistrés ; or cette méthode limite l'interaction et le dynamisme sonore d'une scène, en plus de requérir des banques de sons très lourdes à enregistrer puis à gérer.

Une approche plus générique consiste à synthétiser le son en temps réel à l'aide d'algorithmes : on parle de génération « procédurale » – sont également utilisés, de manière plus ou moins interchangeable, les termes de son « génératif », « algorithmique », « interactif ». C'est la méthode retenue au cours de ces travaux, dont l'objectif est le développement d'un outil de synthèse utilisable par des *sound designers*. La difficulté tient alors au contrôle du son généré : il faut pouvoir associer la perception d'un son à ses paramètres acoustiques, puis relier ces derniers à des paramètres de synthèse. Ces derniers ne doivent pas être trop nombreux et rester suffisamment intuitifs pour être manipulés sans prendre connaissance du modèle de synthèse utilisé.

Nous présentons ici la mise en œuvre de l'enregistrement de sons réels, puis du processus d'analyse et synthèse menant à la réalisation d'un outil de synthèse de sons environnementaux. Cet outil joue le rôle d'une preuve de concept – et sert d'argument en faveur de la génération procédurale de son ; il est également destiné à prendre place dans une application commercialisée par AudioGaming pour une utilisation en production : *AudioWeather*.

Chapitre 1

Enregistrement d'un corpus audio

1.1 Contexte : prise de son Ambisonic

Dans le cadre de nos travaux, l'utilisation d'une base de données de sons de référence est indispensable : elle permet l'analyse puis la comparaison avec les sons synthétisés. AudioGaming étant une entreprise toute récente, elle ne dispose pas d'une grande base de données audio réutilisable dans le cadre de ces travaux.

Il est possible d'utiliser des bases préexistantes, mais cette solution n'est pas viable à long terme – elle ne permet pas d'obtenir tous les sons nécessaires et des problèmes de droits peuvent se poser. Il faut donc pouvoir disposer de nos propres enregistrements. Le matériel d'enregistrement retenu consiste essentiellement en un micro Soundfield SPS200 (voir figure 1.2). Ce micro exploite la technologie Ambisonic, qui a l'avantage de permettre une prise de son multicanal très simplement.

Un assemblage tétraédrique de capsules permet de décomposer le champ de pression selon 3 directions (largeur, profondeur, hauteur) et une 1 composante omnidirectionnelle. Ce format de 4 composantes – le *format B*, qui est à l'heure actuel un standard de fait – est une excellente solution pour le stockage : en effectuant une simple com-

Figure 1.1 – Sonosax SX-R4

Figure 1.2 – Micro Soundfield SPS200

binaison linéaire, on peut par la suite simuler la prise de son depuis n'importe quel dispositif situé au même endroit que le micro Soundfield, pointant vers n'importe quelle direction. [Amb, 2007] rassemble l'essentiel de la documentation sur le sujet. En pratique toutes les configurations de rediffusion ne sont pas idéales, mais pour un système de deux à cinq hauts parleurs dans le plan le rendu est excellent.

Le matériel d'enregistrement comporte également un enregistreur numérique 4 pistes Sonosax SX-R4 (voir figure 1.1).

1.2 Processus d'enregistrement Ambisonic

La prise de son s'est déroulée sur trois semaines, dans plusieurs environnements et situations caractéristiques : gare, aéroport, campagne, ville, orages, temps calme, etc.

En plus des heures d'enregistrement qui constituent un embryon de base audio, c'est surtout la procédure d'enregistrement qui a été mise en place : une chaîne de traitement logicielle sur un PC dédié. Elle couvre la conversion du format natif du micro (4 voies) en *format B*. Le tout est stocké et surtout documenté, avec une fiche détaillant le contexte de la prise et une description des évènements sonores qui ont lieu.

Toutes les données sont numérisées avec une fréquence d'échantillonnage de 48 ou 96 kHz, et quantifiés à 24 bits – ce qui doit permettre d'assurer aux enregistrements une certaine pérennité : on pourra par la suite effectuer des traitements numériques sur le signal sans que la différence de qualité avec le fichier original soit perceptible.

Chapitre 2

Analyse de sons environnementaux

L'un des principaux objectifs de nos travaux consiste à mettre en place un processus d'analyse du son, et de l'éprouver sur des sons d'ambiance environnementaux. On définit ces derniers d'une manière très générale comme les sons autre que la parole ou la musique ; en pratique, ces travaux se consacreront essentiellement sur les sons de vent. Toutefois nous garderons en mémoire la possibilité de réadapter cette étude pour d'autres types de sons.

Nous exposerons d'abord les motivations d'une telle analyse, et l'état de l'art dans ce domaine ; le cas de la classification sera ensuite plus particulièrement détaillé.

2.1 Motivation : *data mining*

Le volume mondial de données numériques augmente exponentiellement : il double tous les trente mois environ [1].
L'analyse et la classification prennent donc une importance grandissante dans la gestion de ces données – des outils d'aperçu, de *thumbnailing*, existent depuis plusieurs années déjà pour l'image ; l'audio ne doit pas faire pas exception à la règle.

Un son « au format CD » comporte pas moins de 44100 échantillons stéréo par seconde ; l'objectif est de se ramener à une « représentation » comportant quelques dizaines d'indices – mille fois moins ! – tout en conservant le maximum d'information utile. L'étape d'analyse débute donc généralement par une réduction ; il ne s'agit pas de transformation ni de compression comme peut le faire un codec (FLAC, MP3...), plutôt de description : on souhaite résumer un signal en une série de descripteurs (les *features*) la plus réduite possible.
Dans notre cas, l'objectif est d'utiliser de tels descripteurs pour classer de manière efficace les sons d'ambiance : parmi chaque *type* (« vent », « pluie », « trafic », etc.), on souhaite distinguer des groupes correspondant chacun à un *ressenti* perceptif (vent « sifflant », pluie « froide ») La détection du type constitue une perspective d'exploration proche du sujet, mais qui dépasse le cadre des travaux présentés ici.

1. *Cisco Visual Networking Index : Forecast and Methodology, 2009-2014*

2.2 État de l'art – Descripteurs

2.2.1 Réduction de dimensionnalité

Il n'existe pas – encore ! – de descripteur idéal, plusieurs modèles coexistent donc.
On peut distinguer d'une part les descripteurs qui découlent directement du signal : étendue de la forme d'onde, centroïdes spectraux, etc. ; d'autre part, les descripteurs qui exploitent les caractéristiques de l'appareil auditif humain [2]. Il peut s'agir d'utiliser des échelles de fréquences non linéaires (coefficients MFC – *Mel-frequency Cepstral*), des échelles d'amplitude logarithmiques ; sont également exploités certains indices psychoacoustiques (sonie, rugosité, etc.) Des descripteurs classiques de la parole ont recours aux modélisations physiques du processus de production du son : coefficients autorégressifs (LPC – *Linear Predictive Coefficients*).

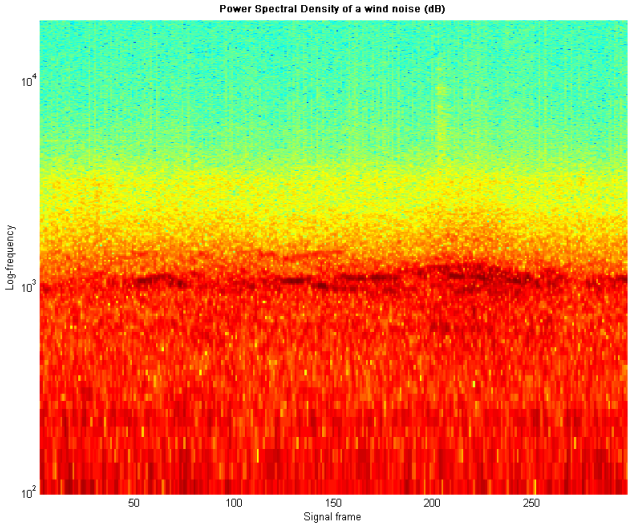
2.2.2 Descripteurs MPEG-7

Ces descripteurs sont créés à partir des données audio brutes, mais n'en font pas partie – il s'agit de métadonnées (« les données à propos des données »). La difficulté réside dans le fait que ces métadonnées englobent aussi bien les paroles du morceau ou sa partition s'il y en a que la variance du signal. Dans un effort louable de conciliation, un standard de description a été créé : il s'agit du MPEG-7, dont la partie 4 consacrée à l'audio a été publiée en 2002 ; les descripteurs sont présentés dans [Peeters, 2004].
Voir fig. 2.1 un exemple d'utilisation pour l'extraction de pics spectraux : le descripteur comporte 24*298 points ; à comparer aux 1024*298 points du spectrogramme, cela n'en représente plus que 2,3 % ; et 5% des données initiales (3 secondes de son échantillonné à 48000 Hz). Or, les pics spectraux sont clairement mis en avant par le descripteur, contrairement à la forme d'onde temporelle ou même à la visualisation par spectrogramme.

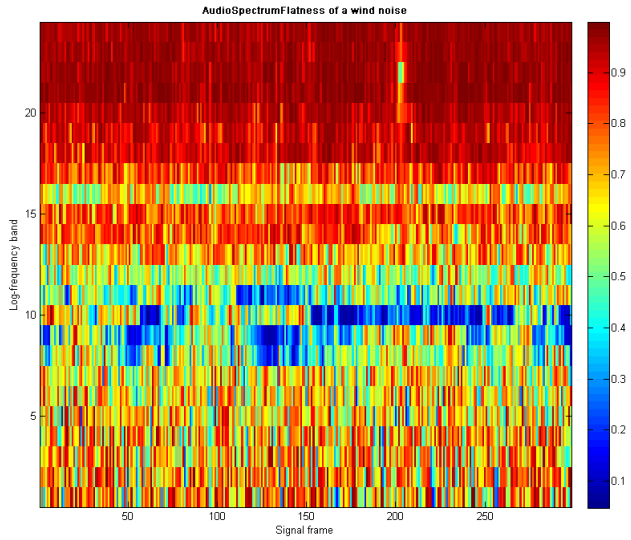
La littérature ne manque pas concernant l'analyse par descripteurs de signaux musicaux ou de parole : des comparaisons sont réalisées entre descripteurs dans [Mayer et al., 2009], et entre descripteurs MPEG et MFCC dans [Kim and Sikora, 2004].

Elle est plus rare pour les sons environnementaux ; ceux-ci constituent un centre d'intérêt relativement récent. [Al-Zhrani and AlQahtani, 2010] par exemple les utilisent à des fins de reconnaissance d'environnement, et en fait [Mitrovic et al., 2007] montrent même par une étude statistiques qu'ils sont plutôt mieux adaptés pour ce type de sons que les traditionnels MFCC, étant moins redondants tout en contenant la plupart des informations données par les descripteurs traditionnels (sonie, *zero-crossing*, etc.).

2. Un usage typique en est fait dans le cadre projet SOMeJB (*SOM-enhanced Jukebox*, ou « Jukebox amélioré par cartes autoadaptatives »), qui applique une série de transformations à un morceau de musique pour le réduire à un descripteur dit *Rhythm Pattern*, utilisé pour l'identification du genre musical (voir [Lidy and Rauber, 2005]).



(a) Spectrogramme



(b) AudioSpectrumFlatness

FIGURE 2.1 – Application d'un descripteur MPEG-7 sur un son réel (vent). a) : spectrogramme (FFT sur 2048 points). b) : descripteur AudioSpectrumFlatness – plus sa valeur est proche de zéro, plus le signal s'approche d'une sinusoïde pure. Noter les pics spectraux vers 1kHz (10ème bande fréquentielle).

2.3 Analyse par descripteurs

2.3.1 Choix d'un ensemble de descripteurs

Les travaux cités ci-dessus nous incitent à sélectionner en priorité des descripteurs MPEG-7 pour leur efficacité à extraire l'information utile dans des sons environnementaux. De plus, l'utilisation finale de ces analyses en sera facilitée : l'interprétation des différentes valeurs de coefficients MFC est plus difficile au moment de l'association à des paramètres de synthèse. Un descripteur MPEG-7 comme un centroïde spectral peut quant à lui s'interpréter bien plus aisément, par exemple comme une fréquence centrale de filtre passe-bande.

L'ensemble de descripteurs est également à adapter aux sons sur lesquelles ils sont appliqués : pour des sons de vent, les descripteurs liés au temps d'attaque ou aux harmoniques ne font pas sens ; d'autres se révèlent redondants (*AudioPower* et *AudioSpectrumEnvelope*). Enfin, certains sont d'une interprétation difficile (classe *Spectral Basis*).

Le sous-ensemble MPEG7 suivant est retenu :

AudioWaveform L'écart entre la valeur minimale et maximale sur chaque fenêtre

AudioSpectrumEnvelope (ASE) Le spectre de puissance, sur une échelle de fréquence logarithmique

AudioSpectrumCentroid (ASC) Le centre de gravité d'ASE

AudioSpectrumSpread (ASS) L'écart type autour de ASC

AudioSpectrumFlatness (ASF) L'aplatissement du spectre de puissance

SpectralCentroid (SC) La moyenne des fréquences pondérée par la puissance.

Ils sont implémentés à l'aide du *MPEG-7 Audio Reference Software Toolkit*, un code Matlab de référence [3] développé dans le projet Cuidado ([Peeters, 2004]).

Après une série de tests informels nous avons fait le choix d'utiliser également des descripteurs perceptifs, issus du projet PsySound3 [4]. Cet ensemble d'outils Matlab présenté dans [Cabrera et al., 2007] permet d'extraire des indices perceptifs ou psychoacoustiques, nous en avons retenu trois :

Roughness Une estimation de la rugosité

Pitch La hauteur perçue

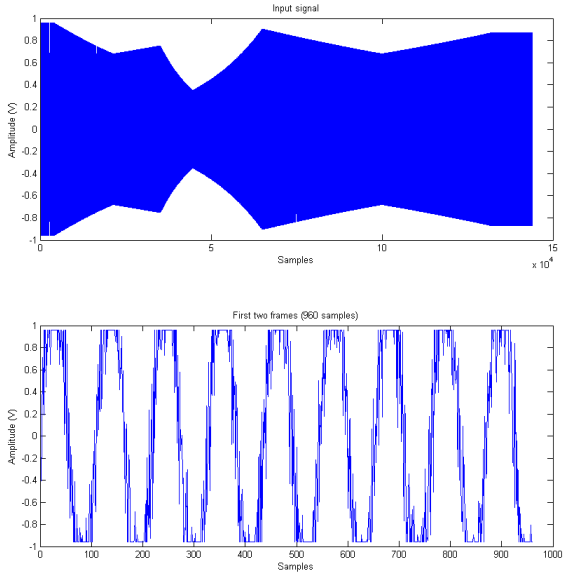
Pitch Strength La « sensation » de hauteur (une sinusoïde pure étant plus sensible qu'une bande de bruit étroite).

Pour faciliter la synthèse, nous avons également décidé d'extraire les deux premiers coefficients LPC à des fins de synthèse, sans les intégrer à l'analyse – nous pouvons en effet réutiliser ces coefficients dans un filtre simulant la forme spectrale du son analysé. Le *zero-crossing* (passage par zéro de la forme d'onde) et les trois premiers coefficients MFC sont également inclus mais à titre expérimental uniquement – ils ne serviront pas

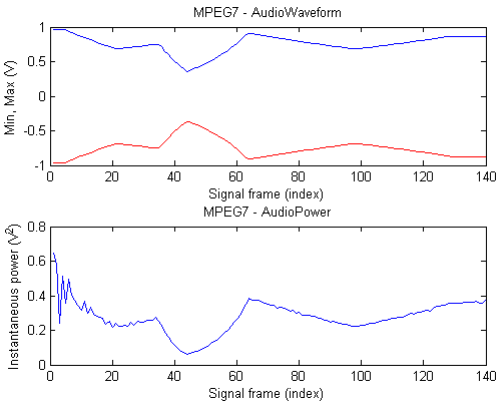
3. L'ensemble de ces ressources est maintenu par Michael Casey et hébergé par le Goldsmiths College : http://mpeg7.doc.gold.ac.uk/

4. Disponible sur http://psysound.wikidot.com

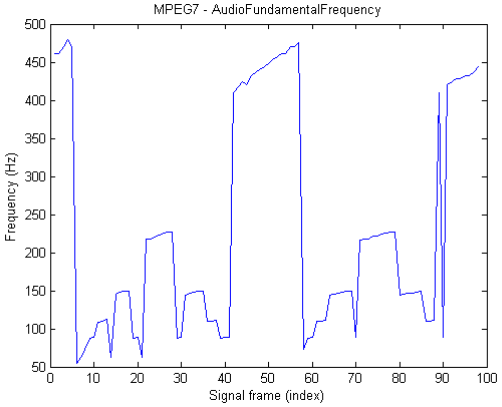
au processus final.
Certains de ces descripteurs sont multidimensionnels ; des tests expérimentaux nous ont permis de déterminer que c'est la médiane plutôt que la moyenne ou la variance qui en constitue le meilleur résumé. Notre ensemble comporte donc 9 descripteurs scalaires ; ils sont appliqués à un signal complexe figures 2.2 et 2.3.



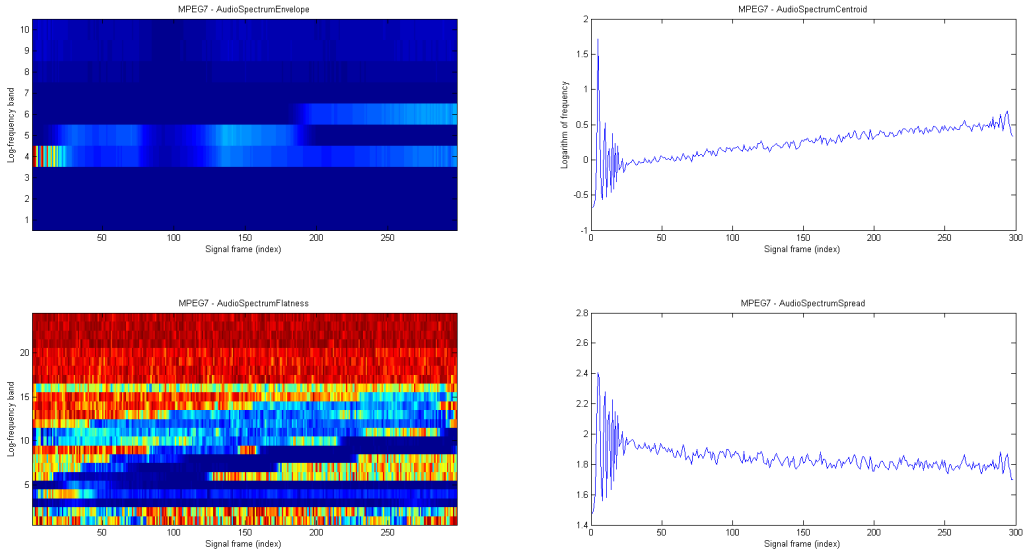
(a) Signal d'entrée (au-dessus ; zoom sur les 2 premières frames en-dessous) composé d'une sinusoïde à 440Hz, de deux sweeps entre (440, 1340) et (1340,3000) et enfin de bruit blanc uniforme au-delà de 4000Hz



(b) Classe *Basic*

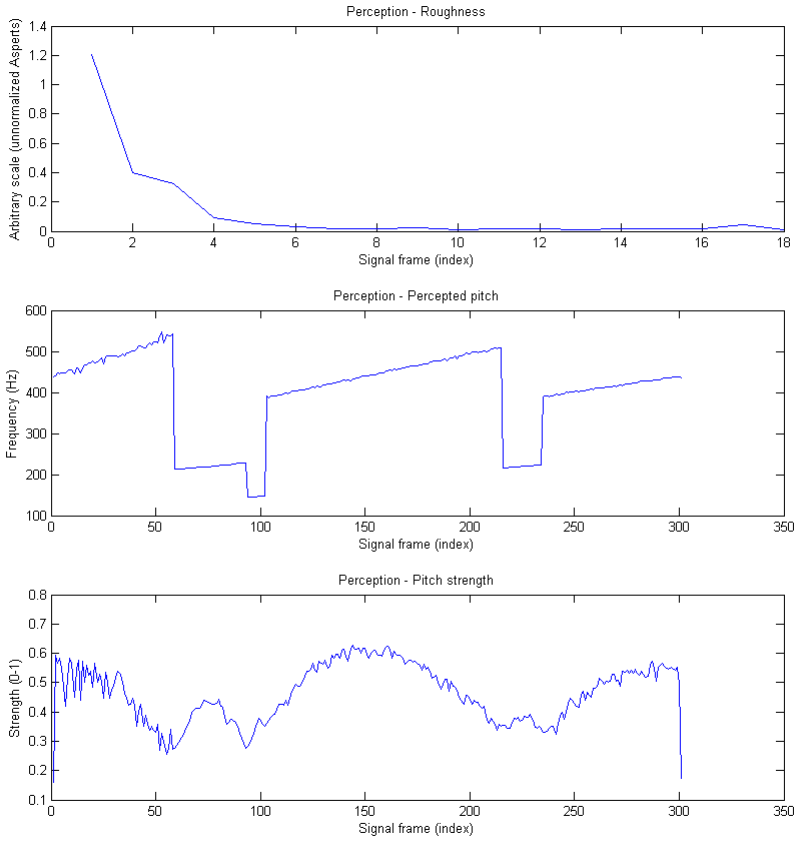


(c) Classe *Signal Parameters*



(d) Classe *Audio Spectrum*

FIGURE 2.2 – Descripteurs MPEG



(a) De haut en bas – *Roughness*, *Pitch*, *Pitch strength*

FIGURE 2.3 – Descripteurs perceptifs

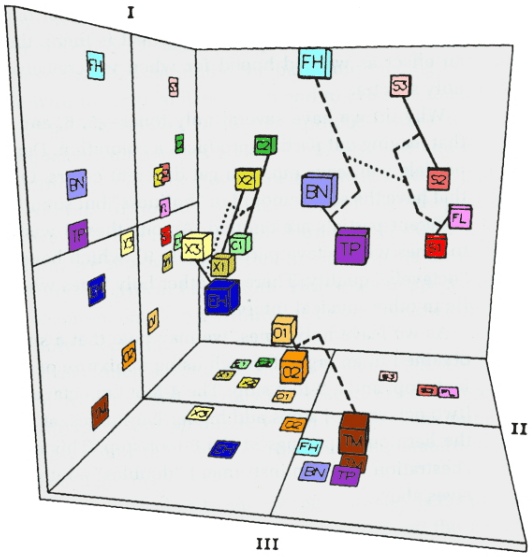


FIGURE 2.4 – Espace de timbre de Grey. Le premier axe est la distribution de l'énergie spectrale, le deuxième intègre la fluctuation spectrale, et le troisième concerne la distribution temporelle des attaques.

2.4 Classification

2.4.1 Espace de perception

Une fois les sons représentés sous une forme compacte, on cherche à relier la perception qu'en ont les auditeurs avec les données de description objective extraites.

Le point de départ de cette classification est la notion d'*espace de timbre* (fig. 2.4) décrit dans [Grey, 1977]. Initialement utilisée pour la description de timbres musicaux, sa méthode consiste à classer des sons instrumentaux par similarité ; une méthode de mise à l'échelle multidimensionnelle (MDS) permet de classer les sons selon la façon dont ils sont perçus, distinguant par exemple les cordes et les cuivres.

Ce modèle a été depuis raffiné par [McAdams et al., 1995] mais les résultats sont souvent similaires, c'est-à-dire que les trois premières dimensions perceptives sont associées respectivement aux caractéristiques spectrales du son (par exemple, le centroïde spectral), à son attaque (forme de l'enveloppe), et à l'irrégularité de son enveloppe spectrale. Or, ces dimensions qui restent pertinentes pour l'analyse de sons instrumentaux ne conviennent plus forcément aux sons environnementaux, comme démontré dans [Keller and Berger, 2001].

2.4.2 Dimensions perceptives

L'idée d'espace de timbre perceptif reste intéressante : [Gygi et al., 2007] par exemple le fait avec succès. C'est encore plus valable en ayant en vue la synthèse de tels sons : cet espace devrait faciliter un contrôle de la synthèse basé sur la perception, en se déplaçant dans l'espace de timbre, plutôt qu'en manipulant des paramètres bas ni-

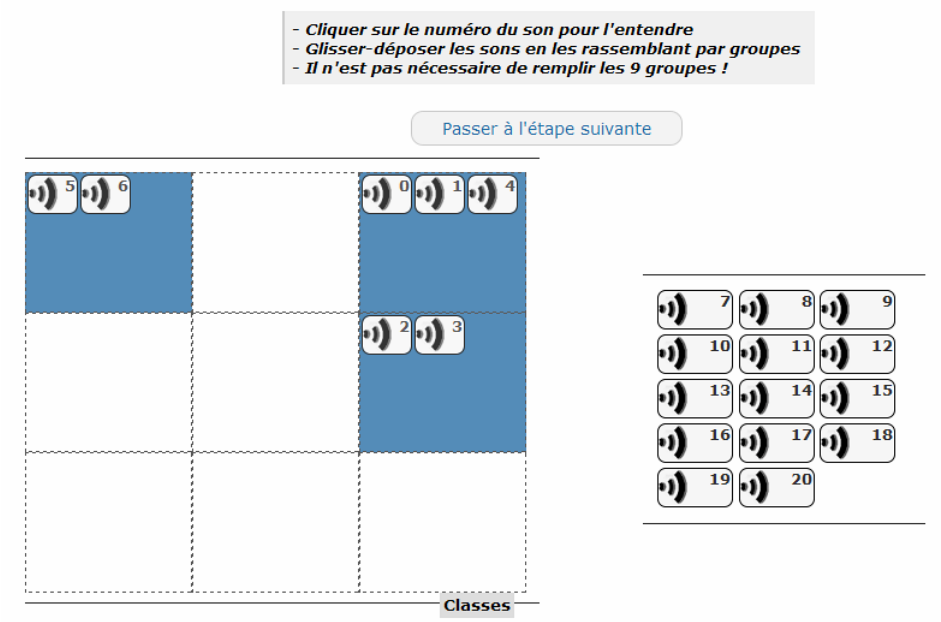


FIGURE 2.5 – Interface de l'expérience perceptive

veaux complexes. Il faut donc pouvoir utiliser des axes adaptés, assemblés à partir des descripteurs choisis plus haut, et représentant des dimensions perceptives.

Cette idée a été mise en œuvre sur les sons environnementaux par Catherine Guastavino dans ses travaux [Guastavino, 2007] et [Guastavino and Katz, 2004] : ses expériences consistent à la fois en des catégorisations par similarité et une verbalisation libre de l'auditeur.

C'est ce type d'approche que nous avons retenu.

2.5 Expérience perceptive

2.5.1 Mise en place

Le protocole retenu, et donc inspiré des travaux de Catherine Guastavino, vise à minimiser l'à priori : aucune indication préalable n'est fournie au sujet, les suggestions sont limitées au maximum ; ceci afin de favoriser la diversité des résultats plutôt que la précision de la classification.
Suivant cette logique, aucune limite de temps n'est fixée, l'expérience n'est pas chronométrée ; le sujet n'est pas supervisé pendant l'expérience, car celle-ci consiste en une application web facilement diffusée (fig. 2.5).
La version finale de cette expérience soumet aux auditeurs vingt et un sons de vent réel provenant d'enregistrement réalisés plus tôt et de base de données audio ; tous les extraits sont normalisés en amplitude et d'une durée limitée à trois secondes.

2.5.2 Résultats obtenus

Onze classifications de sujet sont retenues ; elles comportent en moyenne un peu moins de 4 classes – soit des groupes d'un peu plus de 5 sons chacun. Les descriptions varient d'adjectifs simples à des textes plus élaborés et nous permettent d'extraire des attributs perceptifs qui fassent sens pour désigner des sons de vent : « blizzard », « sifflant », « chaud », etc.
On constate par ailleurs que les descriptions sont aussi bien qualificatives, portant sur la qualité sonore des extraits, que sur les sources ayant produits ce son ; cette constatation a déjà été faite par [Castellengo,] et est particulièrement pertinente pour les sons environnementaux. Les résultats de classification varient donc d'un sujet à l'autre, ne s'attardent pas toujours sur les mêmes critères : ils ne permettent pas d'extraire simplement des classes de sons, ce qui n'est pas inattendu.
Deux résultats sont exploitables : d'une part, certains sons se voient assignés des descriptions très proches par un à deux tiers des sujets, ils peuvent donc faire office de sons « iconiques ». Surtout, les mêmes champs lexicaux sont utilisés : c'est à partir d'eux que l'on définit les attributs perceptifs utilisés par la suite. Pour l'analyse, on retient trois attributs ; les causes qui y semblent associées sont également indiquées, ils sont le résultat d'écoutes et de verbalisations informelles et n'ont qu'une simple valeur descriptive. Il s'agit de :

Rafales La variation d'énergie

Couleur Le spectre du vent ; il pourrait en fait s'agir de la *température*, sauf que les synonymes de « froid », « chaud » souvent utilisés ne semblent pas exclusifs !

Sifflantes Présence de pics spectraux

Les données de l'expérience d'une part et les descripteurs d'autre part sont traitées à l'aide d'un outil [5] de *data mining* décrit dans [Hall et al., 2009]. Il nous permet de recouper les données et d'identifier le sous-ensemble de descripteurs le plus pertinent pour chaque attribut perceptif (fig. 2.6), c'est-à-dire le plus parcimonieux. [Witten et al., 2011] détaille plusieurs techniques à ce sujet ; dans notre cas, étant donné le faible nombre d'instances (21 sons), une recherche exhaustif est effectuée.
Pour chacun de ces attributs, on extrait donc un descripteur « perceptif » : il permet de classifier les sons objectivement, selon des données perceptives. La classification selon l'attribut *Rafales* de notre ensemble de test est donné fig. 2.7 – il est composé des descripteurs *PitchStrength*, *AudioSpectrumEnvelope*, *Rugosité* (par ordre d'importance).

2.6 Conclusion et perspectives

Nous avons relié des descripteurs du signal acoustique avec des attributs perceptifs donnés par des auditeurs. Suite à notre processus d'analyse, la notion de *rafales*, par exemple, peut être représentée objectivement par une série de descripteurs : nous pouvons ainsi qualifier les sons à l'aide d'un algorithme. À court terme, ces résultats sont utilisés pour créer un modèle de synthèse du vent qui sera discuté plus loin (Chap. 3).
À plus long terme, nous espérons d'une part améliorer le processus grâce à des expériences perceptives à plus grande échelle et d'autres descripteurs, plus proches

5. Weka, développé par l'université de Waikato : http ://www.cs.waikato.ac.nz/ml/weka/

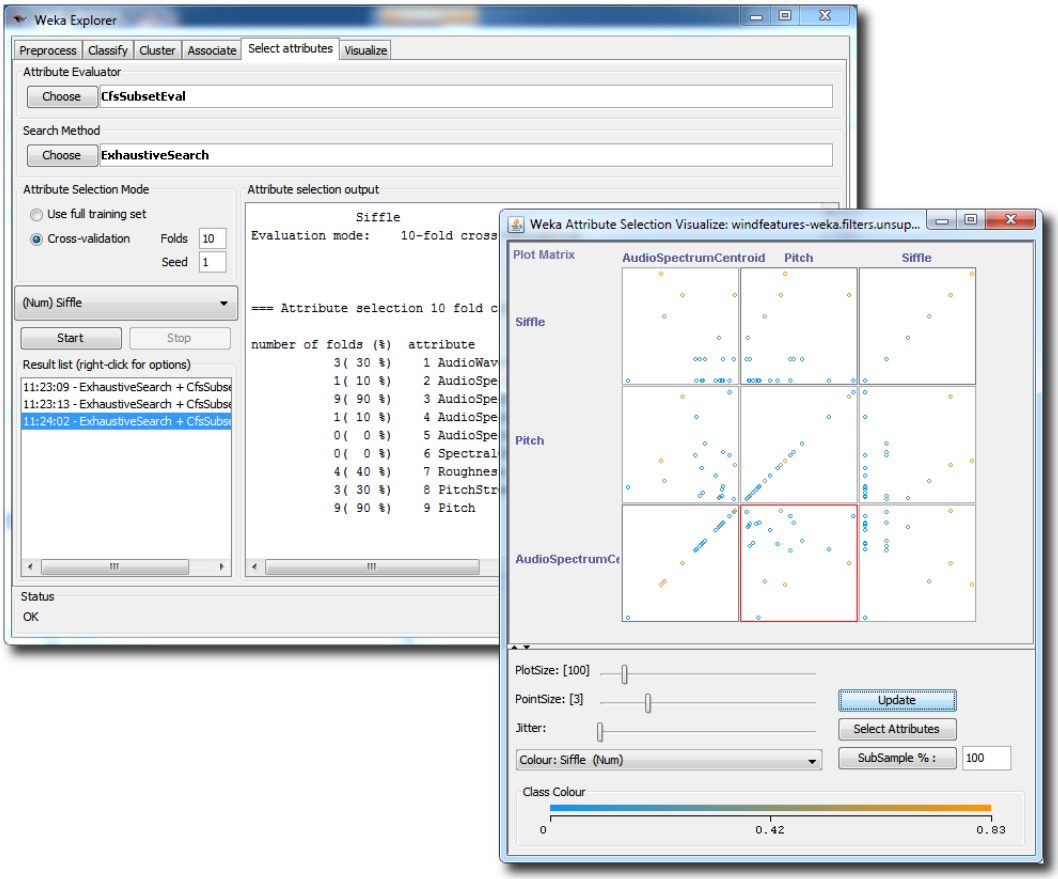


FIGURE 2.6 – Extraction de l'ensemble de descripteurs le plus pertinent pour l'attribut « sifflante » à l'aide *Weka*

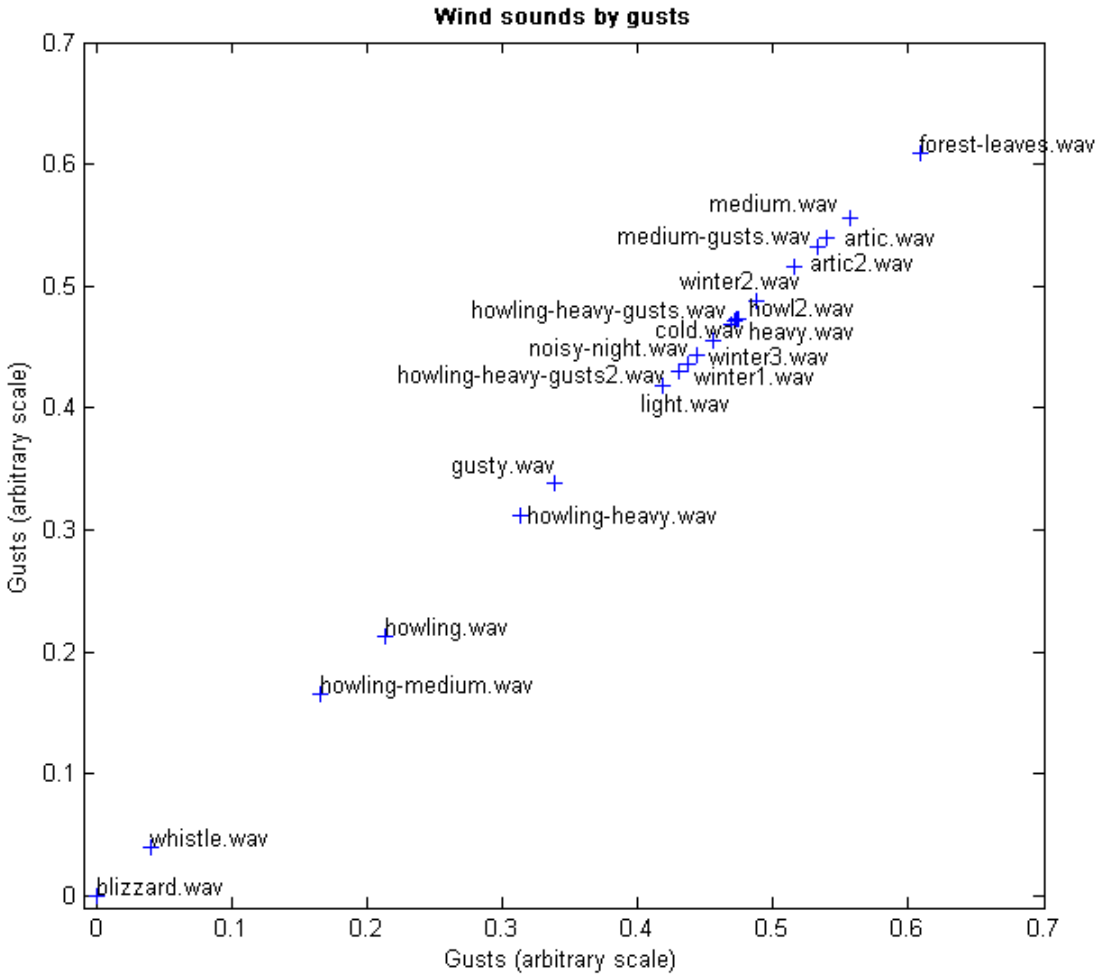


Figure 2.7 – Projection des sons selon le descripteur « rafales ».

de la synthèse. En effet, certains effets sonores sont ignorés pour l'instant, faute de descripteur adéquat (les bruissements, par exemple). D'autre part, nous envisageons d'adapter ce processus à d'autres types de sons environnementaux (pluie, etc.) : le processus d'analyse pourra ainsi être automatisé et mis en œuvre en aval d'un système de segmentation chargé de détecter le type.
Enfin, plusieurs applications peuvent être envisagées lorsque le processus d'analyse est couplé à un système de synthèse ; ces possibilités seront discutées en section 4.

Chapitre 3

Synthèse de sons environnementaux

3.1 L'audio dans le jeu vidéo

3.1.1 Historique : apparition

Nous l'avons évoqué dans l'introduction, la synthèse de son n'est que très peu utilisée dans le jeu vidéo. En fait, il serait plus exact d'écrire que la synthèse a été progressivement abandonnée ; jusqu'en 1990, le rejeu d'échantillons était impossible compte tenu de leur place prise en mémoire. Les effets sonores et la musique étaient donc en très grande majorité synthétisés à la volée – certaines consoles, ainsi que les premières cartes son *SoundBlaster* (à ce moment le standard de fait sur PC), comportent des chips audio de synthèse (FM en particulier).
Le coût global de la mémoire diminuant constamment, les jeux apparus au début des années 90 (la console SNES est sortie en 1990) firent de plus en plus appel à des musiques composées à l'aide de *trackers*, permettant d'assembler des échantillons préenregistrés sur une grille de temps, et de les rejouer à la volée ; de la même manière, des échantillons sont rejoués sur des évènements particuliers du joueur pour figurer des effets sonores.
La démocratisation du CD a permis la lecture de son de très haute qualité, directement depuis le support, et a constitué la dernière grosse innovation dans le domaine ; les progrès réalisés depuis ayant été plutôt quantitatifs – plus de canaux de sortie, des échantillons de meilleure qualité, etc.

3.1.2 État de l'art

La synthèse sonore, explorée dans d'autres domaines d'application (musique, travaux acoustiques) réapparaît depuis peu comme une alternative au rejeu de sons préenregistrés ; le récent ouvrage de Andy Farnell ([Farnell, 2010]) rassemble et résume les raisons de ce changement de paradigme.
Le principal inconvénient de cette méthode concerne les effets sonores. La nécessité d'enregistrer toutes les interactions possibles entre les éléments d'un environnement virtuel pose très rapidement problème : le nombre d'interactions possibles entre éléments devient très vite ingérable.

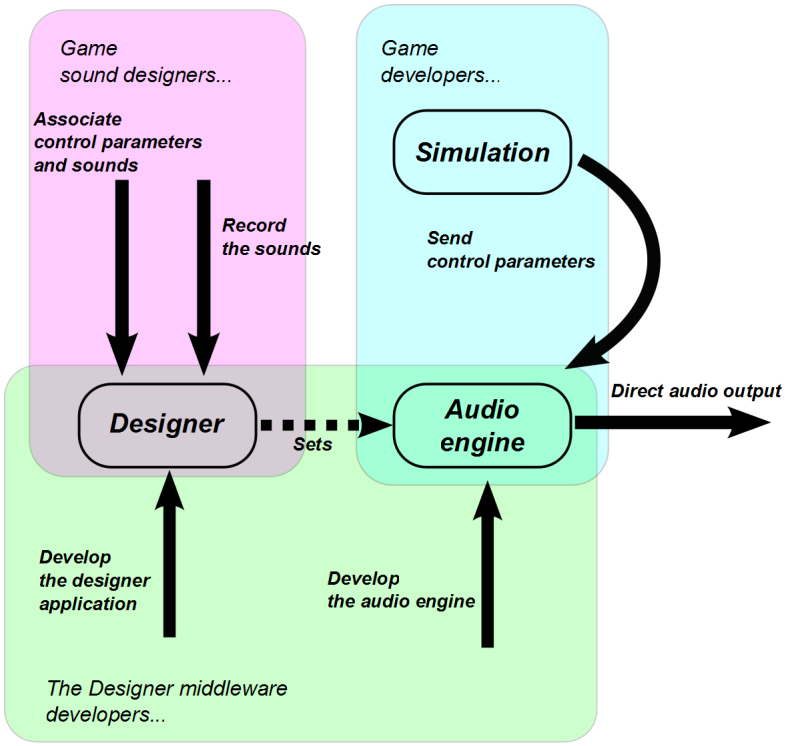


FIGURE 3.1 – Cas d'utilisation classique de la création audio pour le jeu vidéo.

D'autre part, l'interactivité s'améliorant dans la plupart des autres domaines (graphismes, simulation physique), l'audio ne peut pas toujours se limiter au comportement figé qu'imposent les échantillons. Pour limiter l'impression d'un environnement audio statique, des logiciels « *Designer* » (citons *Wwise*, *FMOD Designer*, *Miles*...) sont consacrés à la gestion des échantillons, pour les associer à des actions précises. Des effets y sont ajoutés : des modifications de l'échantillon, comme une distorsion, mais également de l'environnement, avec de l'écho, de la spatialisation, etc. Le tout doit ensuite être intégré dans la simulation finale – le plus souvent sous la forme d'un moteur audio monolithique (fig. 3.1). Le système complet atteint ainsi une grande complexité, sans pour autant permettre d'interactivité poussée – les sons enregistrés constituant la base du système.

À titre d'exemple cité dans [Brandon, 2008], un jeu récent peut consacrer jusqu'à 30% de la capacité totale du support final (DVD) pour stocker l'audio (musique, voix, effets sonores) ; ces données nécessitent la mobilisation d'une équipe de plusieurs personnes, travaillant à temps plein pendant plusieurs mois, pour enregistrer la base de données audio puis effectuer la postproduction et l'intégration dans le projet final.

La synthèse audio pour le jeu vidéo constitue donc une alternative attendue, et un domaine de recherche très actif. Les travaux de [Lloyd et al., 2011] constituent un exemple très récent, pour les sons d'impact, d'un modèle implémenté sur console de jeu avec les contraintes d'optimisation afférentes. Cécile Picard, dans sa thèse

[Picard-Limpens, 2009], propose d'exploiter les données existantes de jeux vidéo – textures, modèles 3D – pour faciliter la synthèse de son en temps réel. *Wwise* cité plus haut propose depuis peu la génération procédurale de sons de vent et d'impact.

3.2 Synthèse de sons environnementaux

3.2.1 Contraintes de performances, solutions existantes

Les applications de simulation qui ne sont pas destinées à fonctionner sur un support matériel dédié ont des contraintes de performances très fortes, qui ont d'ailleurs limité jusqu'à maintenant l'utilisation massive d'audio procédural. Pour un jeu vidéo commercial actuel, [Brandon, 2008] rapporte que le processus audio complet se voit alloué entre 2 et 5% du temps processeur, et jusqu'à 10% de la mémoire disponible – c'est-à-dire entre 20 et 64 Mo selon la plate-forme, pour les PC et consoles de salon. Notre système de synthèse doit donc être créé en prenant en compte ces contraintes et les différentes plate-formes ciblées.
Cela limite en particulier le recours à la modélisation physique en temps réel, dont les résultats sont convaincants, mais majoritairement très lourds en temps de calcul – [Zhang et al., 2005] par exemple doit recourir à des calculs sur processeur graphique afin d'obtenir un résultat satisfaisant en temps réel ; [Dobashi et al., 2003] présente un système de synthèse de sons de vent en temps réel... Nécessitant des calculs préliminaires de plusieurs heures !
D'où l'idée introduite par Perry Cook ([Cook, 1997]) de modèles de synthèse « physiquement informés », c'est-à-dire simplement inspirés du processus physique de création du son. C'est également l'approche de [Verron, 2010], qui reprend les travaux d'Andy Farnell cités plus haut et insiste sur le contrôle « haut niveau » – l'utilisation de paramètres intuitifs.

Un démonstrateur existe déjà à AudioGaming (fig. 3.2), à titre de preuve de concept. Il consiste en une chaîne d'éléments « bas niveaux » de traitement de signal (générateurs et filtres) contrôlés en temps réel. Les paramètres de ces éléments sont liés à des paramètres de haut niveau arbitrairement choisis (température du vent, etc.) qui peuvent être modifiés en temps réel. Les résultats ne sont pas suffisamment variés ni facilement contrôlable par un utilisateur lambda. Il s'agit donc d'utiliser un système de contrôle issu de l'analyse perceptive, et si possible d'améliorer les performances.

3.2.2 Modèle général retenu – *AudioWeather*

Le système mis en place par AudioGaming retient l'idée d'un modèle de synthèse sous forme de *patch* : il s'agit toujours de construire une chaîne d'éléments de traitement du signal sonore. Des paramètres perceptifs de haut niveau sont alors *abstraits*, c'est-à-dire qu'ils peuvent alors être utilisés depuis l'extérieur pour piloter le modèle.
Le processus de synthèse se fait donc en deux temps : création du modèle à l'aide d'un

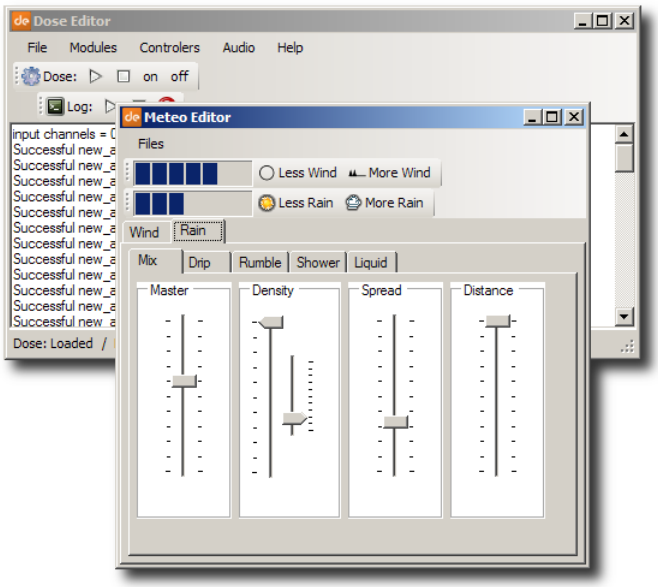


FIGURE 3.2 – Dose, le prototype d'éditeur d'AudioGaming.

langage interprété, puis génération de code compilable. On cumule la simplicité d'un langage interprété et les performances d'un code compilé ; il s'agit d'une démarche similaire à celle du langage FAUST ([Smith, 2011]), bien que les fondements soient assez différents.

Dans nos travaux, le modèle de synthèse de vent (fig. 3.3) est basé sur les résultats de la phase d'analyse ; on retrouve les trois attributs de *rafales*, de *couleur* et de *sifflantes* qui constituent donc nos paramètres perceptifs de haut niveau. Deux autres paramètres moins essentiels ont été ajoutés dans le modèle final : le *muffling*, qui nous permet de gérer les basses fréquences d'une façon proche de celle d'un *sound designer* (par exemple pour gérer un canal basses fréquences), et les *bruissements*, qui n'étaient pas pris en compte dans l'analyse.

Cette architecture nécessite donc un éditeur de paramètres qui permette de relier les paramètres de contrôle utilisés dans la simulation aux paramètres perceptifs de haut niveau utilisés par le modèle de synthèse. Ce logiciel fait office de « *Designer* ».

3.2.3 Implémentation

Les modèles de synthèse consistent donc en des chaînes de traitement de base. Ces unités de base sont écrites en C et sont indépendantes. La synthèse est faite dans le domaine temporel pour des raisons de simplicité : la synthèse par transformée inverse décrite dans [Rodet and Depalle, 1992] et mise en place par Charles Verron aurait nécessité une trop grande durée de développement. Toutefois, cette possibilité n'est pas définitivement écartée (voir section 3.3).
On distingue les unités de signal qui agissent sur les buffers audio, des unités de

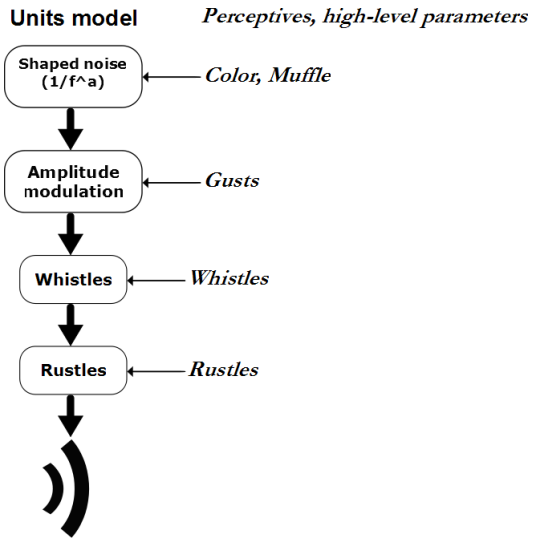


FIGURE 3.3 – Modèle de vent piloté par des paramètres perceptifs

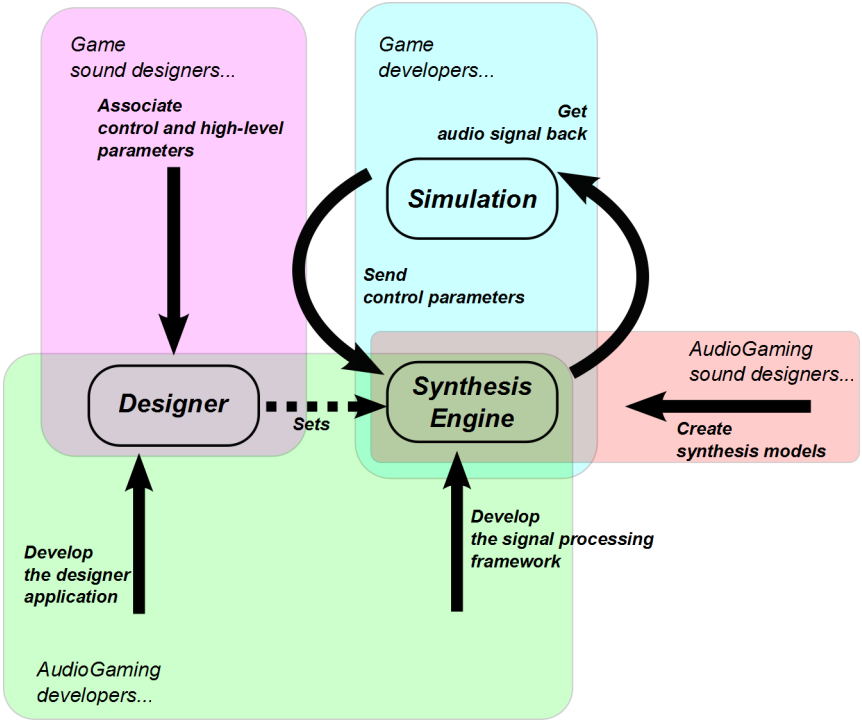


FIGURE 3.4 – Schéma général d'utilisation – AudioWeather

contrôle qui agissent sur les paramètres. Les premières doivent fonctionner suffisamment vite pour fournir au moteur audio les échantillons requis, à la fréquence d'échantillonnage prévue en aval ; les unités de contrôle peuvent requérir des mises à jour moins fréquentes.
L'assemblage des unités peut se faire en C ou C++, mais également en Python grâce à un *wrapper* permettant d'appeler les unités en C : l'idée réside dans le fait qu'un langage interprété permet de réaliser des prototypes plus rapidement. Une fois le modèle réalisé, le code correspondant est généré directement depuis Python : il s'agit d'une classe C++ du modèle, utilisable depuis n'importe quelle application C++ – voire d'autres langages depuis lesquels la bibliothèque dynamique est acceptée. L'application est par exemple utilisable depuis le moteur de jeu *Unity3D* : l'appel à la bibliothèque AudioWeather se fait depuis le langage C# (fig. 3.5).

3.3 Conclusion et perspectives

Nous avons utilisé les résultats du processus d'analyse pour créer un modèle de synthèse de vent piloté par des paramètres perceptifs de haut niveau.
Le modèle a été implémenté sous forme d'une bibliothèque dynamique (dll) codée en C et C++ ; elle a été intégrée avec succès dans une scène de moteur de jeu (*Unity*) mais également portée sur consoles (PS3, Xbox 360), prouvant l'adaptabilité de l'architecture retenue. Par rapport au modèle du prototype original, le résultat perceptif est similaire, le contrôle meilleur, et le temps de calcul divisé par un facteur 10 environ sur le PC de développement. Cette dernière remarque valide l'utilisation d'une telle méthode dans une simulation temps réel où les performances sont critiques.
La synthèse du son est susceptible de bénéficier de nombreuses améliorations : tout d'abord, une analyse détaillée permet d'affirmer que les performances peuvent encore être améliorées par une optimisation spécifique à chaque plate-forme cible. Ensuite, l'utilisation d'unités de traitement du signal fonctionnant dans le domaine fréquentiel laisse entrevoir des effets de synthèse intéressants et peu coûteux en calculs – les travaux de Charles Verron prouvent la faisabilité d'une telle approche.
Enfin, à titre de validation, nous envisageons de réaliser des tests d'écoute sur les sons synthétisés, afin d'estimer formellement la qualité de notre synthétiseur par rapport aux enregistrements réalisés au début de ces travaux.

FIGURE 3.5 – Intégration dans une application Unity – AudioWeather

Chapitre 4

Conclusion générale

Résultats

Nous avons présenté le développement du logiciel de synthèse de sons environnementaux commencé depuis quatre mois au sein d'AudioGaming ; si les travaux décrits ici se concentrent plus particulièrement sur la synthèse de sons de vent, le processus décrit peut s'appliquer à tous types de sons.
Après une première phase d'enregistrement de données de référence, nous avons exploré les différentes études portant sur l'analyse de sons environnementaux. La question de la perception du timbre, particulièrement importante dans le cas de sons non musicaux, a d'abord été abordée par la mise en place d'une expérience perceptive. Est venue ensuite la problématique de relier les données perceptives de cette expérience avec le signal acoustique – nous avons utilisé dans ce but des outils de description de contenu.
Un modèle de synthèse des sons de vent a été développé sur la base de ces études. Il a été implémenté à l'aide d'outils de programmation mis en place à AudioGaming, qui ont permis de remplir les exigences de performance requise pour une utilisation en temps réel dans un contexte de jeu vidéo. Le modèle est contrôlé par des paramètres perceptifs de haut niveau qui facilitent son utilisation par les *sound designers*, sans la nécessité de connaître les détails du modèle lui-même.

Perspectives

En plus des améliorations suggérées pour chacun des processus d'analyse et de synthèse, c'est l'ensemble de ces deux éléments qui constitue un support de recherche à long terme.
Le premier axe de ces recherches concerne le couplage des processus de synthèse et d'analyse, qui est à l'heure actuelle réalisé « à la main ».
En effet, la perspective de pouvoir automatiser le système d'analyse/synthèse permettrait la paramétrisation automatique . Un tel système pourra ainsi recevoir en entrée des sons enregistrés de référence et, par une méthode itérative de *machine learning*,

trouver l'ensemble de paramètres optimal pour le reproduire par la synthèse. Sans même automatiser entièrement le processus, mais en se limitant à un pré-réglage des paramètres du modèle, la tâche de *sound design* s'en trouverait grandement facilitée. Surtout, la mise en place d'une telle automatisation permettra de vérifier simplement si le domaine des sons synthétisés couvre bien l'ensemble du domaine des sons perçus du même type ; c'est-à-dire la capacité de notre synthétiseur de vent à synthétiser toutes les nuances que l'auditeur est capable de distinguer.
Le second axe de recherche concerne la spatialisation. La synthèse actuelle se limite au moment de l'écriture de ce mémoire (août 2011) à un son monocanal. La solution la plus simple consiste à générer un signal pour chaque canal ; mais des possibilités plus sophistiquées permettraient de simuler facilement la spatialisation tout en requérant peu de calculs supplémentaires. En fait, la possibilité évoquée dans la section 3.3 d'utiliser des unités de synthèse dans le domaine fréquentiel permettrait une mise en œuvre peu coûteuse en temps de calcul des techniques de filtrage adaptées.

Bibliographie

[Amb, 2007] (2007). The ambisonic network.

[Al-Zhrani and AlQahtani, 2010] Al-Zhrani, S. and AlQahtani, M. (2010). Audio Environment Recognition using Zero Crossing Features and MPEG-7 Descriptors. *Journal of Computer Science*, 6 :1262–1266.

[Brandon, 2008] Brandon, A. (2008). Next-gen audio square-off : Playstation 3 vs. xbox 360.

[Cabrera et al., 2007] Cabrera, D., Ferguson, S., and Schubert, E. (2007). Psysound3 - software for acoustical and psychoacoustical analysis of sound recordings. In Scavone, G. P., editor, *Proceedings of the 13th International Conference on Auditory Display (ICAD2007)*, pages 356–363, Montreal, Canada. Schulich School of Music, McGill University, Schulich School of Music, McGill University.

[Castellengo,] Castellengo, M. Caractérisation perceptive des sons. .

[Cook, 1997] Cook, P. R. (1997). Physically informed sonic modeling (phism) : Synthesis of percussive sounds. *Computer Music J.*, 21(3) :38–49.

[Dobashi et al., 2003] Dobashi, Y., Yamamoto, T., and Nishita, T. (2003). Real-time rendering of aerodynamic sound using sound textures based on computational fluid dynamics. In *Proceedings of ACM SIGGRAPH Aug. 03*, pages 732–740.

[Farnell, 2010] Farnell, A. (2010). *Designing Sound*. Applied Scientific Press Limited.

[Grey, 1977] Grey, J. (1977). Multidimensional perceptual scaling of musical timbres. *Acoustical Society of America Journal*, 61 :1270–1277.

[Guastavino, 2007] Guastavino, C. (2007). Categorization of environmental sounds. *Canadian Journal of Experimental Psychology*, 60(1) :54–63.

[Guastavino and Katz, 2004] Guastavino, C. and Katz, B. (2004). Perceptual evaluation of multi-dimensional spatial audio reproduction. *Journal of the Acoustical Society of America*, 116 :1105–1115.

[Gygi et al., 2007] Gygi, B., Kidd, G. R., and Watson, C. S. (2007). Similarity and categorization of environmental sounds. *Perception and Psychophysics*, 69 :839–855.

[Hall et al., 2009] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software : an update. *SIGKDD Explor. Newsl.*, 11 :10–18.

[Keller and Berger, 2001] Keller, D. and Berger, J. (2001). Everyday sounds : synthesis parameters and perceptual correlates. In *Proceedings of the VIII Brazilian Symposium of Computer Music held in Fortaleza*.

[Kim and Sikora, 2004] Kim, H. and Sikora, T. (2004). Comparison of mpeg-7 audio spectrum projection features and mfcc applied to speaker recognition, sound classification and audio segmentation. In *Advances in Kernel Methods – Support Vector Learning*, pages 925–928. MIT Press.

[Lidy and Rauber, 2005] Lidy, T. and Rauber, A. (2005). Evaluation of feature extractors and psycho-acoustic transformations for music genre classification.

[Lloyd et al., 2011] Lloyd, D. B., Raghuvanshi, N., and Govindaraju, N. K. (2011). Sound synthesis for impact sounds in video games. In *I3D '11 Symposium on Interactive 3D Graphics and Games*.

[Mayer et al., 2009] Mayer, R., Frank, J., and Rauber, A. (2009). Analytic comparison of audio feature sets using self-organising maps. In *WEMIS 2009 - Workshop on Exploring Musical Information Spaces*, pages 62–67, Alicante, Spain. University of Alicante. talk : WEMIS 2009 - Workshop on Exploring Musical Information Spaces, Korfu ; 2009-10-01 – 2009-10-02.

[McAdams et al., 1995] McAdams, S., Winsberg, S., Donnadieu, S., De Soete, G., and Krimphoff, J. (1995). Perceptual scaling of synthesized musical timbres : common dimensions, specificities, and latent subject classes. *Psychological Research*, 58(58) :177–192.

[Mitrovic et al., 2007] Mitrovic, D., Zeppelzauer, M., and Eidenberger, H. (2007). Analysis of the data quality of audio descriptions of environmental sounds. *Journal of Digital Information Management*, 5 :48–55.

[Peeters, 2004] Peeters, G. (2004). A large set of audio features for sound description (similarity and classification) in the cuidado project. Technical report, IRCAM.

[Picard-Limpens, 2009] Picard-Limpens, C. (2009). *Expressive Sound Synthesis for Animation*. PhD thesis, Nice-Sophia Antipolis.

[Rodet and Depalle, 1992] Rodet, X. and Depalle, P. (1992). Spectral envelopes and inverse fft synthesis. In *Proceedings of the 93rd Audio Engineering Society Convention*.

[Smith, 2011] Smith, J. (2011). Signal processing in faust.

[Verron, 2010] Verron, C. (2010). *Synthèse immersive de sons d'environnement*. PhD thesis, Aix-Marseille 1.

[Witten et al., 2011] Witten, I. H., Frank, E., and Hall, M. A. (2011). *Data Mining : Practical Machine Learning Tools and Techniques, Third Edition*. Morgan Kaufmann.

[Zhang et al., 2005] Zhang, Q., Ye, L., and Pan, Z. (2005). Physically-based sound synthesis on gpus. In *ICEC*, pages 328–333.