UNIVERSITÉ DE SHERBROOKE
Faculté de génie
Département de génie électrique et de génie informatique

Réalisation d'un système de substitution sensorielle de la vision vers l'audition

Mémoire de maîtrise
Spécialité : génie électrique

Damien LESCAL

Jury : Jean ROUAT - Directeur (UdeS)
Stéphane MOLOTCHNIKOFF - Examinateur externe (UdeM)
Jérémie Voix - Examinateur externe (ÉTS)
Philippe MABILLEAU Rapporteur (UdeS)

Sherbrooke (Québec) Canada Août 2012

RÉSUMÉ

Ce projet de recherche a été mené dans le cadre du groupe de recherche NECOTIS (NEurosciences COmputationnelles et Traitement Intelligent du Signal). Ce groupe de recherche agit principalement dans le domaine du traitement de l'image et de l'audio grâce à des méthodes de traitement de signal bio-inspirées. Différentes applications ont été développées en reconnaissance de la parole, dans la séparation de sources sonores ou encore en reconnaissance d'images.

Bien qu'ils existent depuis plus de quarante ans, les systèmes d'aide aux personnes atteintes de déficiences visuelles, que cela soit des prothèses visuelles (invasif) ou des système de substitution sensorielle (non invasif), n'ont pas percé dans le milieu du handicap. Il serait difficile d'imputer cet état de fait à des limitations technologiques : depuis les premières approches, les prothèses visuelles ou les systèmes de substitution sensorielle n'ont cessé de se perfectionner et de se diversifier. Toutefois, si la question de savoir comment transmettre le signal est bien documentée, la question de savoir quel signal transmettre a été plus rarement abordée.

Différents systèmes ont été développés mais le plus impressionnant est le récit des utilisateurs de tels systèmes. Ainsi, il fait plaisir de lire que l'artiste Neil Harbisson, qui ne voit aucune couleur, explique comment une caméra attachée à se tête lui permet d'entendre des couleurs et ainsi de pouvoir peindre [Montandon, 2004]. Un autre exemple tout aussi impressionnant, la scientifique Wanda Díaz-Merced, qui travaille pour xSonify, explique comment elle analyse différentes données en les encodant de façon sonore [Feder, 2012].

C'est dans ce cadre que ce projet de substitution sensorielle de la vision vers l'audition a été développé. En effet, nous avons utilisé le traitement de signal bio-inspiré afin d'extraire différentes caractéristiques représentatives de la vision. De plus, nous avons essayé de générer un son agréable à l'oreille et représentatif de l'environnement dans lequel évolue la personne. Ce projet a donc davantage été axé sur la nature du signal transmis à la personne ayant des déficiences visuelles.

Mots-clés : Substitution sensorielle, non-voyant, réseau de neurones, fonction de transfert relative à la tête (HRTF, *Head Related Transfer Function*)...

REMERCIEMENTS

En préambule à ce mémoire, je souhaiterais adresser mes remerciements les plus sincères aux personnes qui m'ont apporté leur aide et qui ont contribué à l'élaboration de ce projet ainsi qu'à l'aboutissement de ces deux formidables années universitaires.

Je tiens à remercier sincèrement Monsieur Jean Rouat, qui, en tant que Directeur de recherche, s'est toujours montré à l'écoute et très disponible tout au long de la réalisation de ce mémoire. L'inspiration, l'aide et le temps qu'il a bien voulu me consacrer ont été indispensables à la réalisation de ce mémoire.

Mes remerciements vont également à Monsieur Jérémie Voix (Titulaire de la Chaire de Recherche Industrielle en Technologies Intra-Auriculaire Sonomax-ÉTS, CRITIAS) pour sa générosité et le temps qu'il m'a consacré malgré ses charges académiques et professionnelles. L'accès qu'il nous a donné au laboratoire ICAR - Infrastructure Commune en Acoustique pour la Recherche ÉTS-IRSST a été indispensable pour les mesures des fonctions de transfert relatives à la tête. Je tiens aussi à remercier les étudiants du groupe de recherche CRITIAS ainsi que les techniciens de l'ÉTS qui m'ont permis de réaliser mes mesures dans un cadre accueillant.

J'exprime ma gratitude à tous les consultants et internautes rencontrés lors des recherches effectuées et qui ont accepté de répondre à mes questions avec gentillesse.

Enfin, un merci tout particulier à mes parents pour leur soutien et leur contribution mais aussi à mes proches et amis, qui m'ont toujours soutenu et encouragé au cours de ces deux dernières années.

À toutes et à tous, merci.

TABLE DES MATIÈRES

LISTE DES FIGURES

LISTE DES TABLEAUX

LISTE DES ACRONYMES

Acronyme	**Définition**
HRTF	*Head Related Transfer Function* : Fonction de transfert Relative à la tête
ILD	*Interaural Loudness Difference* : Différence Interaurale de Niveau sonore
ITD	*Interaural Time Difference* : Différence Interaurale de Temps
ARP	*Auditory Research Platform* : Plateforme pour la recherche auditive

CHAPITRE 1

INTRODUCTION

1.1 Mise en contexte et problématique

Depuis plus de quarante ans, différents types de systèmes d'aide aux personnes atteintes de déficiences visuelles, que cela soit des prothèses visuelles (invasif) ou des système de substitution sensorielle (non invasif) ont vu le jour. L'évolution technologique a permis de concevoir des implants de plus en plus petits avec de meilleurs résultats. Cette même évolution a permis aux systèmes de substitution sensorielle de devenir de plus en plus portables (introduction de systèmes sans fil) et aussi de plus en plus complexes. En effet, tandis que les premiers systèmes reproduisaient de façon tactile la scène visuelle sur de petites portions de peau, les nouveaux systèmes se sont développés autour de matrices d'électrodes avec une meilleure résolution ou bien avec des matrices plus grandes. Un nouveau type de substitution sensorielle a aussi vu le jour : la substitution de la vision par l'audition. Ces nouveaux types de systèmes ont été développés afin de ne plus représenter la scène visuelle par des stimuli tactiles mais afin de représenter cette même scène par des stimuli sonores. Tous ces systèmes de substitution sensorielle permettent d'éviter des interventions chirurgicales lourdes et coûteuses qu'entraînent la pose d'implants.

1.2 Définition du projet de recherche

C'est dans cette lignée de systèmes de substitution sensorielle que notre projet vient se placer. En effet, ce projet vise la mise en oeuvre d'un système complet de substitution sensorielle de la vision vers l'audition. Plus précisément, après acquisition vidéo des images, nous traitons celles-ci afin de récupérer uniquement les informations (parties) importantes de l'image. Ensuite, à partir des informations obtenues par le traitement d'images, nous générons un son représentatif de la scène visuelle qui est transmis au système auditif via des écouteurs.

1.3 Objectifs du projet de recherche

L'objectif de cette recherche est de modéliser et de réaliser un système de substitution sensorielle de la vision vers l'audition. En effet, nous nous appuyons sur la plasticité du cerveau [Kokovay *et al.*, 2008; Pascual-Leone *et al.*, 2005] afin de réaliser un système permettant à des non-voyants la reconnaissance d'objets simples à partir de stimuli sonores. Afin de réaliser ce projet, l'objectif premier est de faire un traitement intelligent de l'image qui prend en considération certaines caractéristiques de la vision. Le deuxième objectif est d'utiliser les éléments extraits de l'image afin de produire un stimulus sonore permettant aux non-voyants de se représenter l'environnement devant eux. Dans un troisième temps, afin de valider rapidement le système complet, nous avons développé celui-ci sous forme d'application pour iPhone/iPad/iPod™. Par la suite, si le modèle réalisé semble valide, il pourrait alors être envisagé de passer à la réalisation pratique d'un appareil utilisant ce système afin de le soumettre à la communauté scientifique pour être testé ainsi qu'aux personnes souffrant de déficience visuelle pour être à nouveau testé puis utilisé. Cet appareil comprend une caméra (iPhone/iPad/iPod™), des écouteurs et deux microphones intégrés aux écouteurs. L'iPhone/iPad/iPod™ joue le rôle d'unité de calcul et d'acquisition : une application développé sur iPhone/iPad/iPod™ permet à la caméra de celui-ci de capturer les images puis de les traiter afin de générer des sons représentants l'image. Les deux microphones intégrés aux écouteurs permettent de capter le son ambiant. Enfin, les écouteurs permettent de transmettre les sons générés par le système à l'utilisateur, mixés (si l'utilisateur le souhaite) avec le son ambiant capturé par les deux microphones externes situés sur les écouteurs.

1.4 Contribution originale

Ce projet de recherche vient s'ajouter dans la liste des systèmes de substitution sensorielle qui ont déjà été développés. Ces systèmes ont prouvé non pas qu'il était possible de remplacer un système sensoriel par un autre mais qu'il était possible d'aider des personnes ayant perdu un sens à retrouver certaines sensations via un autre sens. Ces systèmes viennent aussi conforter la théorie autour de la notion de plasticité du cerveau. La plasticité cérébrale est un terme qui décrit les mécanismes par lesquels le cerveau est capable de se modifier par l'expérience. Ce phénomène intervient durant le développement embryonnaire, l'enfance, la vie adulte et les conditions pathologiques (lésions et maladies). Cette théorie montre que le cerveau est un système dynamique, en perpétuelle reconfiguration.

Ce projet de recherche se différencie des autres systèmes de substitution sensorielle de la vision vers l'audition à quatre niveaux :

1. Au niveau du traitement de l'image : la technique de traitement de l'image nous permet d'accéder aux éléments importants présents dans l'image.
2. Au niveau de la génération de sons : la technique de génération de sons nous permet de générer des sons agréables à l'oreille et qui caractérisent la position de l'élément représenté au niveau horizontal et vertical sans avoir à scanner l'image.
3. Au niveau du système global : le système est développé sous iPhone/iPad/iPod™ avec utilisation d'écouteurs conventionnels.
4. Une autre version a été créée pour interfacer avec des écouteurs personnalisés (ARP de SONOMAX) qui ont l'originalité de permettre d'entendre ce qui se passe dans l'environnement extérieur à l'aide d'un microphone intégré.

En conclusion, le système s'inspire des caractéristiques humaines de la vision (sélection des régions importantes de l'images) et de l'audition (encodage de la position des objets). De plus, le système complet est développé sous iPhone/iPad/iPod™, ce qui le rend très accessible et facilement transportable.

1.5 Plan du document

La suite de ce document est structurée de la façon décrite ci-après.

Dans une première partie (chapitre 2), un état de l'art est présenté. Celui-ci permet de situer le projet de recherche par rapport aux travaux publiés dans le même domaine.

Ensuite, la réalisation du projet est détaillée. Plus précisément, dans un premier temps,le système complet et son fonctionnement sont expliqués (chapitre 3). Dans un second temps, nous détaillons les techniques utilisées afin d'identifier et de localiser les différentes zones importantes présentes dans une scène visuelle (chapitre 4). Par la suite, nous présentons de quelle façon nous générons des sons agréables et localisables (chapitre 5). Finalement, les tests du système et les résultats sont présentés (chapitre 6).

En guise de conclusion sont présentés un sommaire, les contributions et les travaux futurs (chapitre 7).

CHAPITRE 2

État de l'art

2.1 Systèmes de substitution sensorielle

Le cerveau est extrêmement plastique [Kokovay *et al.*, 2008; Pascual-Leone *et al.*, 2005] et il est possible d'utiliser la substitution pour renforcer un système sensoriel par un autre [Bach-Y-Rita *et al.*, 1969]. Il existe deux grandes catégories de systèmes qui permettent de compenser la perte de la vision : des systèmes de substitution de la vision vers le tactile qui convertissent des images en des stimulations tactiles et des systèmes de substitution de la vision vers un signal qui convertissent des images en sons.

2.1.1 Substitution de la vision vers le tactile

De nombreux systèmes de substitution de la vision vers le tactile ont été développés. Dans la majorité de ces systèmes, les images sont obtenues à partir d'une caméra et sont ensuite transformées en vibrations électriques appliquées sur la peau d'une partie du corps (dos, abdomen, langue, doigts, etc). Des recherches ont montré que l'utilisation de tels systèmes pouvait permettre aux utilisateurs d'effectuer différentes tâches : reconnaissance de formes simples [Kaczmarek et Haase, 2003; Sampaio *et al.*, 2001], lecture [Bliss *et al.*, 1970], et localisation [Jansson, 1983]. Un des systèmes ayant apporté des résultats intéressants est le système étudié par Bach-Y-Rita [Bach-Y-Rita, 2003]. Cette étude a montré qu'il était possible d'utiliser la langue comme média de stimulation du cerveau afin de permettre à un non-voyant de se déplacer et de « visualiser » son environnement. Ce modèle a même été breveté par Hogle R. et Lederer S. en décembre 2009. C'est ainsi que, dans certaines salles d'opération, les chirurgiens pourront être assistés par des stimulations nerveuses de la langue afin de mieux percevoir [Akinbiyi *et al.*, 2006]. Cependant, ces systèmes utilisent des parties très sensibles de notre corps et sont donc propices à des problèmes comme l'irritation de la peau ou diverses douleurs. De plus, bien que ces systèmes de substitution aient montré leurs potentiels, ils restent inefficaces pour certains types de cécité comme celle due au diabète (importante cause de perte de la vue), car la maladie entraîne une perte de sensibilité tactile.

2.1.2 Substitution de la vision vers l'audition

L'autre partie de la substitution visuelle est la substitution de la vision vers l'audition. L'audition présente de nombreux avantages : le système auditif humain permet de traiter des sons complexes et changeants rapidement comme la parole même dans des milieux bruités. Les différents systèmes existants se divisent en deux catégories : les systèmes basés sur l'echolocalisation et les systèmes basés sur la conversion d'image en son.

Les systèmes d'echolocalasition sont basés sur le même principe que les sonars. Ces systèmes utilisent un émetteur/récepteur à ultrasons. Le récepteur utilise les règles de télémétrie afin de déterminer la distance entre la source et l'objet. Ensuite, les signaux sont convertis dans des fréquences audibles et transmis à l'utilisateur via des écouteurs. Ces systèmes permettent aux utilisateurs d'avoir une indication de la distance et de la direction d'un objet. Par exemple, la distance peut être codée par la tonalité et la position horizontale par la disparité interaurale. Ces systèmes peuvent être d'une grande aide pour la locomotion et pour guider les mouvements des personnes non-voyantes [Kay, 1964]. Ils peuvent aussi apporter des informations sur la représentation spatiale d'une scène en trois dimensions [Hughes, 2001].

En ce qui concerne les systèmes qui n'utilisent pas l'approche télémétrique, les images d'une caméra sont converties en sons et transmises à l'utilisateur via des écouteurs. Trois systèmes principaux ont déjà été étudiés : vOICe développé par Meijer en 1992 [Meijer, 1992], PSVA (Prosthesis for substitution of vision by audition) développé par Capelle et al en 1998 [Capelle *et al.*, 1998] et l'appareil développé par Cronly-Dillon en 1999 [Cronly-Dillon *et al.*, 1999]. Ces trois systèmes convertissent la position verticale d'un objet lumineux dans l'image vidéo en différentes fréquences : les sons aigus représentent le haut de l'image et les sons graves le bas de l'image. Le système développé par Cronly-Dillon diffère des autres systèmes car il utilise une gamme de fréquences plus limitée qui correspond à des notes musicales. Pour ce qui est de la position horizontale, vOICe et le système développé par Cronly-Dillon scannent l'image de gauche à droite dans le temps. Le PSVA utilise quant à lui les fréquences : la fréquence associée à chaque pixel augmente de la droite vers la gauche et du bas vers le haut. Différentes études de ces systèmes ont montré la possibilité de localiser des objets [Amedi *et al.*, 2007; Auvray *et al.*, 2007; Cronly-Dillon *et al.*, 1999, 2000; Merabet *et al.*, 2009; Pollok *et al.*, 2005; Renier *et al.*, 2005a]. Une autre étude a montré qu'il était possible de recréer des illusions visuelles avec le PSVA [Renier *et al.*, 2005b].

Plus récemment, un autre système a été développé : The Vibe [Hanneton *et al.*, 2010]. Il se rapproche du PSVA car il code l'image dans sa globalité, sans balayage gauche/droite. Il utilise un modèle de rétine qui découpe l'image en plusieurs champs récepteurs (zones concentriques). Il associe un son à chaque champ récepteur et la somme de tous ces sons simples forme un son complexe qui est transmis aux deux oreilles.

2.2 Traitement d'images : approche objet

Si l'homme sait naturellement séparer des objets dans une image, c'est grâce à des connaissances de haut niveau (compréhension des objets et de la scène). Mettre au point des algorithmes de segmentation de haut niveau (chaque région est un objet sémantique) est encore un des thèmes de recherche les plus courants en traitement d'images.

2.2.1 L'attention et le système visuel humain

Ce que l'homme voit est déterminé par ce qu'il est intéressé à voir. L'attention visuelle permet aux personnes de sélectionner les informations les plus importantes suivant la situation en cours. Ainsi, une même image peut être présentée plusieurs fois à la même personne et les éléments retenus à chaque fois seront différents suivant l'état d'esprit de la personne [Chun et Wolfe, 2001]. Aussi, la dualité attention/vision peut être vue de différents points de vue [Wolfe, 2000] et différents éléments (couleurs, orientations, tailles...) sont à prendre en compte pour comprendre le phénomène d'attention visuelle. Ce phénomène d'attention chez l'homme peut être caractérisé d'un point de vue traitement d'images par une approche objet. On entend par approche objet un traitement de l'image qui permet de récupérer uniquement certains éléments (objets) présents dans l'image.

2.2.2 Modèles de traitement d'images adaptés à la vision : approche objet

Alors que le système visuel humain est capable naturellement de différencier les différents objets présents dans une scène visuelle, cette tâche reste très difficile à réaliser pour des systèmes de traitement d'images. Toutefois, de nombreux types de traitement d'images existent et peuvent être comparés [Pantofaru, 2005]. Ces différentes techniques de traitement d'images étudient les images numériques et leurs transformations, dans le but d'améliorer leur qualité ou d'en extraire de l'information. Dans cette grande « famille », la segmentation d'images est une opération qui a pour but de rassembler des pixels entre eux suivant des critères prédéfinis. Les pixels sont ainsi regroupés en régions, qui constituent une partition de l'image. Il peut s'agir par exemple de séparer les objets du fond.

La localisation et la recherche d'objets dans les images reste un domaine de recherche très actif. Des méthodes récentes telles que celle proposée par Kokkinos et Yuille [Kokkinos et Yuille, 2009] trouvent et suivent des objets issus d'images ou de vidéos en utilisant des caractéristiques relatives à la structure des objets. D'autres systèmes ont été développés en se basant sur des modèles de la vision comme HMAX [Riesenhuber et Poggio, 1999] et sur des systèmes hiérarchiques [Pinto *et al.*, 2011; Riesenhuber et Poggio, 1999]. Dans cette lignée, différents systèmes ont essayé de reproduire le traitement d'images qu'exécute l'oeil en prenant en compte l'attention [Frintrop *et al.*, 2007; Itti, 2000; Peters et O'Sullivan, 2003].

D'un point de vue approche objets, la segmentation d'image et la localisation d'objets basés sur des modèles de « *binding* » [Milner, 1974; Von Der Malsburg, 1981] peuvent maintenant être réalisés en temps réel avec des réseaux de neurones à décharges sur des systèmes embarqués [Caron *et al.*, 2011]. D'autres systèmes tels que ceux-ci [Ryan *et al.*, 2009] proposent une segmentation d'image en temps réel avec mise en avant des objets et des régions d'intérêt. L'approche objet implique un processus « top-down » qui utilise des connaissances à priori [de Ladurantaye *et al.*, 2012]. Toutefois, l'architecture à mettre en place pour un système temps réel est assez lourde. Nous avons donc opté pour une stratégie plus légère en utilisant une approche complètement « bottom-up ». Cependant, la conception du système a été faite pour que le processus identifie des zones importantes qui sont détectées pour traiter des objets : la recherche de contours et de textures élevées.

2.3 Localisation sonore

Localiser une source, c'est déterminer sa direction, donc son azimut et sa hauteur (ou élévation), ainsi que la distance à laquelle elle se trouve, donc sa profondeur.

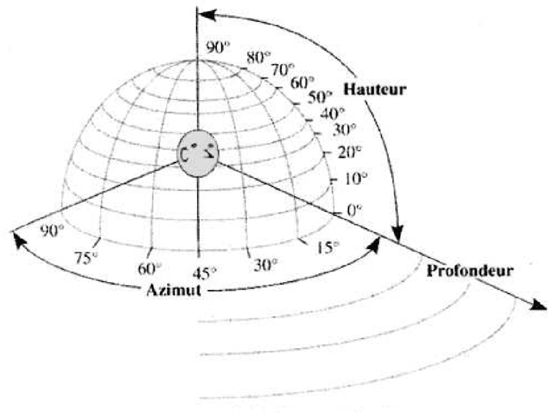


Figure 2.1 La localisation naturelle

L'homme est capable de détecter, d'identifier et de localiser la source des sons autour de lui et même de déterminer approximativement la direction et la distance de chaque source. Lord Rayleigh dans sa « duplex theory » a présenté les fondations de la recherche moderne en localisation sonore [Strutt, 1907]. Blauert quant à lui a défini la localisation comme *the law or rule by which the location of an auditory event (e.g., its direction and distance) is related to a specific attribute or attributes of a sound event* [Blauert, 1983]. L'introduction des notions de la différence interaurale de niveau sonore et de la différence interaurale de temps a beaucoup aidé les recherches dans ce domaine [Brungart *et al.*, 1999; Deatherage et Hirsh, 1959]. De nombreuses études ont été effectuées en localisation statique en utilisant des écouteurs [Blauert, 1983; Wenzel *et al.*, 1993]. Il est bien connu qu'en localisation à l'aide d'écouteurs, le son semble être localisé à l'intérieur de la tête [Junius *et al.*, 2007]. Ce phénomène est connu sous le terme de« lateralization ». D'autres études ont montré que l'extériorisation de son à l'aide d'écouteurs pouvait être réalisée en utilisant des HRTF (*Head-Related Transfer Function*, soit « fonction de transfert relative à la tête ») [Hartmann et Wittenberg, 1996]. Dans la suite de ce document, nous détaillons ces éléments et expliquons comment ils sont utilisés afin de générer la perception spatiale d'un son.

2.3.1 Localisation dans le plan horizontal

La localisation dans le plan horizontal est effectuée par le système auditif en fonction de deux paramètres : la différence interaurale de niveau sonore et la différence interaurale de temps.

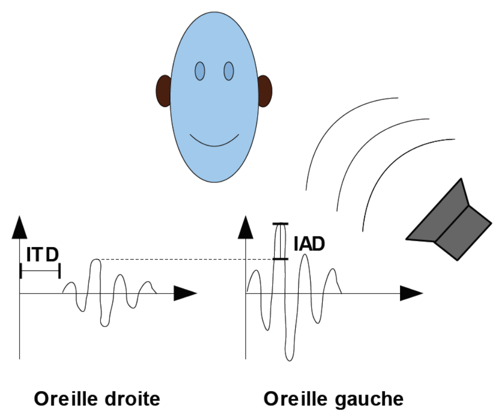


Figure 2.2 Figure explicative des indices binauraux (extrait de la thèse de B.Durette, 2009) : l'amplitude du signal reçue par l'oreille gauche est plus élevée que celle reçue par l'oreille droite(IAD) et le signal arrivera à l'oreille gauche avant d'arriver à l'oreille droite (ITD).

La différence interaurale de niveau sonore (ILD, *Interaural Loudness Difference*)

Suivant la position de la source sonore, le signal va plus ou moins être masqué par la tête. Ce phénomène va entraîner une différence de perception de niveau sonore entre les deux oreilles. En regardant la figure 2.2, on voit que l'amplitude du signal reçue par l'oreille gauche est plus élevée que celle reçue par l'oreille droite (IAD sur la figure). Ce phénomène est observé pour des signaux hautes fréquences ($>$1 200 Hz). La tête, de par sa morphologie, constitue un obstacle pour l'onde sonore. La texture de l'oreille, la forme du pavillon et du canal ainsi que le tympan provoquent des phénomènes de diffraction et d'absorption, d'où un filtrage en fréquence qui dépend de la position de la source sonore. La complexité de ce filtrage est telle qu'il n'existe à ce jour aucune modélisation mathématique vraiment fiable. La figure 2.3 montre, pour une source sonore se déplaçant de 0 à 180°, les différences d'intensité entre les deux oreilles en fréquence obtenues expérimentalement.

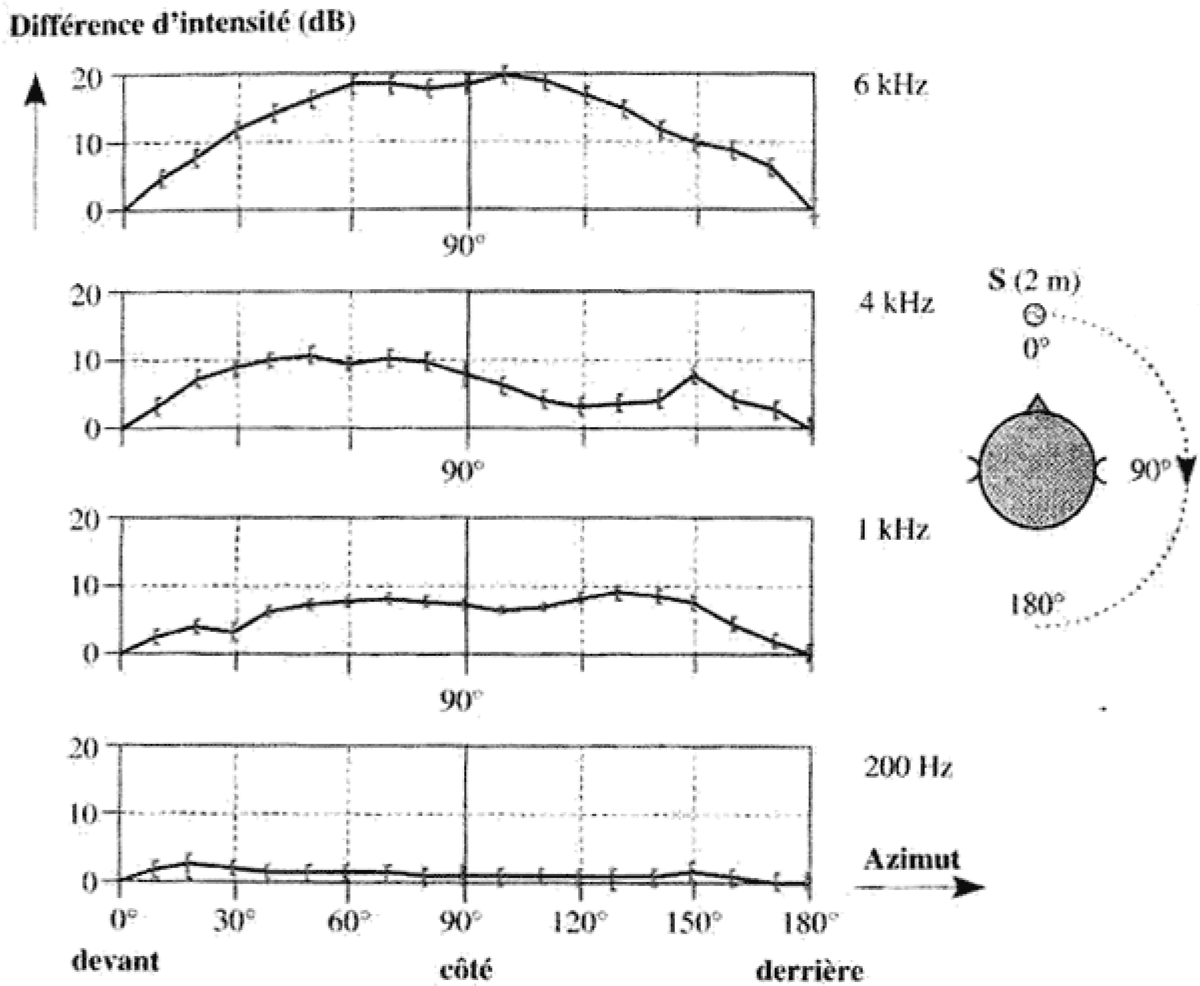


Figure 2.3 Évolution de la différence d'intensité entre les deux oreilles en fonction de la position de la source sonore pour 4 fréquences : 200 Hz, 1 kHz, 4 kHz et 6 kHz. Pour les signaux de 200 Hx, la tête n'influence pas la différence d'intensité tandis que pour les signaux de fréquence égale à 6 kHz, cette différence est maximale (d'après Fedderson, Sandel, Teas et Jefress [Feddersen *et al.*, 1957])

Pour chaque position de la source sonore, la différence d'intensité perçue par chaque oreille est fonction de la fréquence : on parle de fonction de transfert interaurale. Mais les différences d'intensité sont beaucoup plus significatives lorsque la valeur de la longueur d'onde se rapproche des dimensions de la tête, soit, pour les fréquences supérieures à 1 200 Hz.

La différence interaurale de temps (ITD, *Interaural Time Difference*)

En plus de l'ILD, suivant la position de la source sonore, le signal va mettre plus ou moins de temps pour arriver à chacune des oreilles. En effet, en faisant encore référence à la figure 2.2, le signal arrivera à l'oreille gauche avant d'arriver à l'oreille droite (ITD sur la figure). Ce phénomène est observé pour des signaux basses fréquences (<1 200 Hz). Ainsi, si on assimile la tête à une sphère de rayon r, on peut définir la différence de temps d'arrivée des signaux provenant d'une source sonore en fonction de son angle d'incidence.

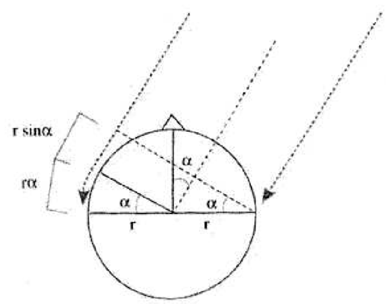


Figure 2.4 Parcours supplémentaire de l'onde de contournement [Blauert, 1983]

On calcule qu'une source située à 90° (localisation latérale extrême) est perçue avec une différence de temps de 0.65 ms, valeur de retard maximum d'une oreille par rapport à l'autre.

2.3.2 Localisation dans le plan vertical

Le système auditif humain permet aussi de localiser la source sur un plan vertical. Cette localisation fait, entre autres, intervenir la forme du lobe de l'oreille et d'autres facteurs. Ceci dit, la localisation d'une source sonore est moins précise dans le plan vertical que dans le plan horizontal. Ce manque de précision est dû à des différences interaurales particulièrement réduites. Ainsi, l'incertitude de localisation atteint 15 à 20° pour une source située au-dessus de la tête.

L'expérimentation décrite à la figure 2.5 met en évidence le rôle déterminant du spectre dans la localisation frontale, au dessus ou arrière. Elle permet d'identifier l'effet de la fréquence centrale d'un signal sur la localisation de celui-ci au niveau vertical. On observe que la direction apparente est imposée par la fréquence centrale de certaines zones spectrales. Les signaux au voisinage de :

- 8 kHz semblent venir du haut

- 1 kHz semblent venir de l'arrière

- 3 kHz et les fréquences basses semblent venir de face.

On peut donc utiliser cette notion pour créer un effet d'élévation sonore en introduisant une accentuation vers 8 kHz.

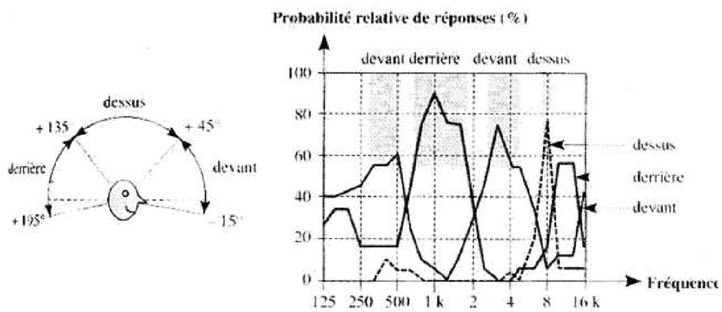


Figure 2.5 Probabilité relative (plus de 50%) de réponses en avant, au dessus et en arrière en fonction de la fréquence centrale du signal. Le signal a été présenté à 20 sujets, une fois devant, une fois derrière et une fois au-dessus (d'après Blauert [Blauert, 1983])

2.3.3 Localisation en profondeur

Trois principaux indices interviennent dans la localisation en profondeur :

- les variations d'intensité

- le rapport du son direct au champ réverbéré

- les variations spectrales typiques

Les variations d'intensité

Lorsqu'on s'éloigne d'une source sonore en champ libre (c'est-à-dire, lorsque le milieu n'est pas trop réverbèrent et que l'on est suffisamment proche de la source sonore), l'intensité diminue de 6 dB à chaque doublement de distance. Diverses expériences ont démontré que des sources sonores familières sont localisées avec plus de précision que des sources inconnues et que dans les deux cas, ces distances sont toujours sous-estimées. L'estimation de la distance dépend également du niveau d'émission de la source. En conséquence, si l'on désire recréer artificiellement une impression de profondeur, une atténuation supérieure à 6 dB (jusqu'à 20 dB) est nécessaire pour donner l'illusion d'un doublement de distance [Blauert, 1983].

Le rapport du son direct au champ réverbéré

Lorsqu'un auditeur s'éloigne d'une source sonore, le rapport du son direct au son réverbéré diminue. Cela vient du fait que l'énergie du son direct décroît de 6 dB à chaque doublement de distance, d'où une même décroissance du rapport champ direct/champ réverbéré. Cette décroissance relative constitue un indice de localisation en profondeur plus représentatif que celui de la seule décroissance du son direct (limité à la distance critique).

Les variations spectrales

La densité spectrale d'un signal sonore varie lors de sa propagation en fonction d'une absorption inégale des fréquences graves et aiguës. Des expériences (Gardner [Gardner, 1968]), ont montré notamment qu'en champ libre, des sons dont le contenu fréquentiel est inférieur à 2 kHz semblent plus éloignés que les sons de fréquence supérieure.

2.3.4 L'espace Acoustique Virtuel

L'Espace Acoustique Virtuel VAS (*Virtual Acoustic Space*) [Schnupp *et al.*, 2011] est une technique dans laquelle les sons présentés à travers des écouteurs semblent venir d'une direction choisie. L'illusion d'une source sonore virtuelle en dehors de la tête de l'auditeur est ainsi créée.

La technique VAS comprend deux étapes :

1. ILD, ITD et les indices spectraux forment la fonction relative à la tête HRTF. Cette fonction définit comment la tête et la forme de l'oreille externe filtrent les sons entrants. La HRTF peut être mesurée en plaçant des microphones miniatures à l'intérieur des oreilles et en enregistrant la réponse impulsionnelle d'une grande quantité de sons présentés à l'oreille de différentes directions dans l'espace [Tianshu *et al.*, 2009]. Vu que la taille de la tête et la forme de l'oreille externe varient d'un individu à l'autre, il est préférable d'individualiser le filtre VAS.

2. Les réponses impulsionnelles de la HRTF sont ensuites converties en un banc de filtres. Ainsi, n'importe quel son peut maintenant être convolué avec un de ces filtres et être joué via des écouteurs. Cela crée la perception d'une source sonore externalisée. Cette approche a des avantages évidents par rapport à la technique « tête de mannequin » : une fois que les différents filtres ont été obtenus, ils peuvent être appliqués à n'importe quelle source sonore désirée.

Un très bon exemple de ce principe extrait du livre de Jan Schnupp, Israel Nelken et Andrew King [Schnupp *et al.*, 2011] est présenté sur le site web [Schnupp, 2011].

CHAPITRE 3

Système global et prototypes

3.1 Présentation du système global

Dans cette section, nous présentons le système global et son fonctionnement.

3.1.1 Schéma bloc du système global

Le signal d'entrée de notre système est une image capturée par une caméra et les signaux de sortie sont des sons représentants les objets présents dans l'image d'entrée. Plus précisément, voici le schéma bloc du système global :

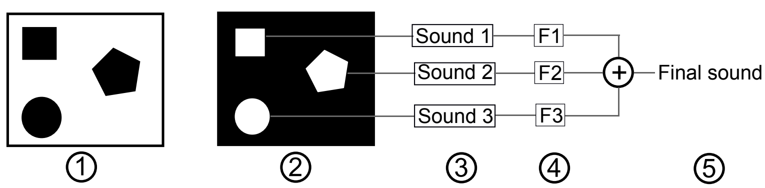


Figure 3.1 Schéma bloc du système global : la section 1 représente l'image capturée, la section 2 représente l'image traitée, la section 3 représente la génération de sons, la section 4 représente le filtrage du son par la technique Espace Acoustique Virtuel (VAS, *Virtual Acoustic Space*) et la section 5 représente le son final.

3.1.2 Fonctionnement du système global

Voici les explications détaillants les différentes étapes de notre système :

1. **Image capturée** : l'image est capturée par une caméra, chaque pixel de l'image est associé à un neurone dans le réseau.
2. **Traitement d'image** : l'image est traitée de façon à ce que seules les zones considérées comme importantes soient prises en compte par le système (section 4.3).
3. **Génération de sons** : un son est associé à chaque zone de l'image (section 5.1).
4. **Filtrage du son** : chaque son est filtré suivant la zone à laquelle il est associé (section 5.3).

5. **Son final** : tous les sons sont mixés puis présentés aux oreilles de l'utilisateur.

3.2 Présentation des prototypes

Dans cette section, nous présentons les prototypes réalisés afin de tester notre système. Le premier prototype a été réalisé afin de tester le bon fonctionnement du système global et aussi afin de valider la localisabilité des sons produits par notre système. En ce qui concerne le deuxième prototype, il introduit l'utilisation des écouteurs ARP (*Auditory Research Platform*) qui permette de reproduire les sons ambiants.

3.2.1 Prototype de base

Dans un premier temps, afin de valider notre système, nous avons implémenté notre système sur iPhone/iPad/iPod™ puis nous avons testé ce système avec des écouteurs simples. Cette première étape a été réalisée afin de tester la localisabilité des sons produits. La génération de sons à partir de l'image capturée est effectué par l'iPhone/iPad/iPod™. Pour cela, nous avons créé une application pour iPhone/iPad/iPod™. Une fois cette application lancée, l'image de la caméra de l'iPhone/iPad/iPod™ est capturée. Nous traitons ensuite cette image par l'algorithme détaillé dans la section 4.3. Ce traitement nous permet de mettre en évidence les zones importantes de l'image. Une fois ces zones identifiées, des sons correspondants à ces zones sont présentés via des écouteurs simples branchés à l'iPhone/iPad/iPod™ (section 5.1). Ces sons sont filtrés afin de les rendre localisables dans l'espace (section 5.2). Voici le schéma bloc du prototype de base (figure 3.2) :

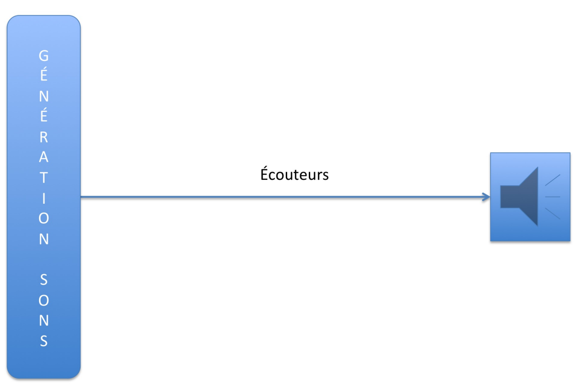


Figure 3.2 Schéma bloc du prototype de base.

Ce premier prototype nous a permis très rapidement de valider notre système : les sons générés étaient localisables.

3.2.2 Prototype avec écouteurs SONOMAX

Dans un second temps, nous souhaitions avoir un système permettant aux utilisateurs de ne pas être coupés de leur environnement sonore. Pour cela, nous voulions reproduire les sons ambiants. Nous avons donc utilisé, grâce à une collaboration avec le Professeur Jérémie Voix de l'ÉTS , des écouteurs ARP développés par SONOMAX. Ces écouteurs ont été développés pour les besoins de la chaire CRITIAS et sont utilisés pour le développement d'une plateforme pour la recherche auditive (figure 3.3).

Figure 3.3 Écouteurs ARP développés par SONOMAX [Sculptedeers, 2012].

Les écouteurs ARP [Critias, 2012], en plus d'être simples et très performants, sont faits sur mesure et le prototype utilisé par CRITIAS possède quatre microphones (deux par oreille). Un microphone est placé à l'intérieur de l'oreille et l'autre à l'extérieur. Le fait que ces écouteurs soient moulés est un avantage majeur car cela permet le positionnement précis des microphones et des haut-parleurs. Dans notre cas, nous utilisons uniquement le microphone externe afin de capturer les sons ambiants pour ensuite les transmettre à l'utilisateur. Le schéma bloc du prototype est présenté à la figure 3.4.

Toute la partie génération de sons à partir de l'image est effectuée par un iPhone/iPad/iPod™. Un boitier, relié d'un côté à l'iPhone/iPad/iPod™ et de l'autre côté aux écouteurs sonomax, permet à l'utilisateur de régler le volume et d'allumer ou d'éteindre le système de génération de sons. Ainsi, l'utilisateur aura constamment le son de l'environnement autour de lui, mais également s'il le souhaite, notre système de génération de sons. Ce petit plus permet aux utilisateurs de ne pas être coupés de l'environnement sonore extérieur (figure 3.5).

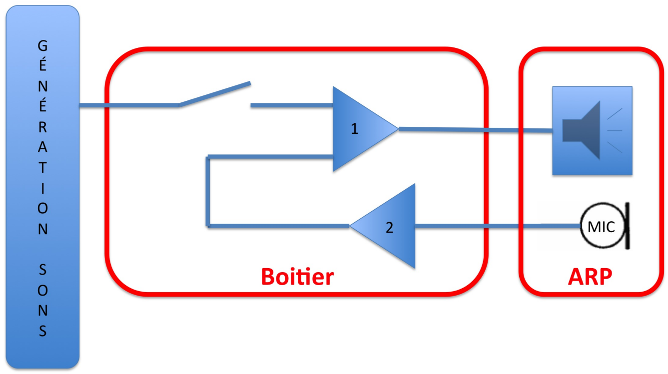


Figure 3.4 Schéma bloc du prototype avec écouteurs SONOMAX : la génération de sons à partir de l'image capturée est effectuée par l'iPhone/iPad/iPod^{TM}. Le boitier comprend deux étages d'amplification et un interrupteur. Cet interrupteur permet de couper le système de génération de sons. L'étage d'amplification à gain fixe (2) permet de mettre à niveau les sons capturés par les microphones des écouteurs ARP. L'étage d'amplification à gain variable et de mixage (1) permet de mélanger les deux signaux et de régler le volume de ceux-ci. Les sons sont transmis via les haut-parleurs des écouteurs ARP.

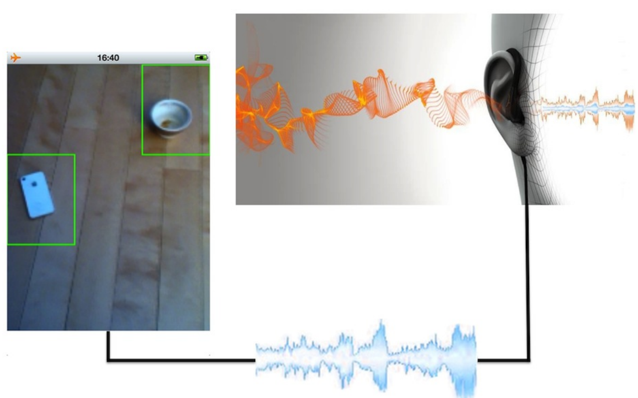


Figure 3.5 Présentation du système global : les sons ambiants (en orange) sont mixés aux sons générés par le système (en bleu) et transmis à l'utilisateur par l'intermédiaire des écouteurs ARP [SONOMAX, 2012].

CHAPITRE 4

Localisation de zones importantes

4.1 Modèle de mise en valeur des zones importantes

Dans cette section, nous décrivons le modèle de réseau de neurones artificiels que nous avons utilisés afin d'identifier les zones importantes dans l'image. Ce travail avait commencé en 2005 par Jean Rouat (Rouat, Rapport interne NECOTIS, 2005) et a été combiné avec les travaux de Louis-Charles Caron [Rouat *et al.*, June 2011].

Ce réseau a été décrit dans l'article présenté à la conférence ISSPA (*Information Sciences, Signal Processing and their Applications*) en juillet 2012. Cet article présente le modèle neurophysiologique sur lequel nous nous sommes appuyés ainsi que le modèle simplifié que nous avons implémenté (l'article se trouve à l'annexe A). Nous détaillons brièvement les principes de cet algorithme dans les trois sections qui suivent.

4.1.1 Fonctionnement du réseau de neurones

Dans ce réseau de neurones, un neurone a une variable d'état et une valeur de pixel. Une synapse est caractérisée par un poids qui dépend de la différence de la valeur des pixels des deux neurones qu'elle connecte. Un algorithme met à jour itérativement l'état des neurones en fonction de leur état actuel, le poids des synapses et de l'état de leurs voisins. Le but de cet algorithme est de différencier les régions homogènes de l'image de celles avec des contours denses. Ces régions sont considérées comme des objets qui seront encodés en sons dans les étapes suivantes du système de substitution sensorielle.

4.1.2 Initialisation des poids de connection

Dans ce réseau, chaque pixel d'une image d'entrée est représentée par un neurone. Les synapses connectent chaque neurone à ses huit neurones voisins. L'état de chaque neurone est initialisé à un. Le poids de la synapse qui connecte deux neurones est calculé en fonction de la différence de la valeur des pixels associés aux neurones de la façon suivante :

$$s_{ij}.w = s_{ji}.w = f\left(abs\left(n_i.p - n_j.p\right)\right), \tag{4.1}$$

où $s_{ij}.w$ est le poids de la synapse qui connecte les neurones n_i et n_j, $f()$ est une fonction non-linéaire possible et $n_i.p$ est la valeur du pixel n_i. La valeur des indices i et j est comprise entre 0 et le nombre de pixels dans l'image. Ainsi, pour chaque neurone indicé i, les valeurs de j correspondront aux indices des 8 voisins du neurone i. Le rôle de la fonction $f()$ est de créer une forte connexion entre les neurones qui ont une valeur de pixel similaire et une faible connexion entre ceux qui ont une valeur de pixel assez différente. La fonction $f()$ doit être définie de telle sorte qu'elle renvoie des valeurs comprises entre 0 et 1. Dans notre cas, nous avons défini $f()$ de la façon suivante :

$$f\left(abs\left(n_i.p - n_j.p\right)\right) = \frac{MAX - abs\left(n_i.p - n_j.p\right)}{MAX}, \tag{4.2}$$

où MAX est la valeur de pixel maximale possible.

4.1.3 Algorithme itératif

Une itération de l'algorithme parcourt la liste des neurones et met à jour leurs états. La mise à jour à l'itération k est calculée de la façon suivante :

$$n_i.s\left(k\right) = \frac{n_i.s\left(k-1\right) + \sum_j n_j.s\left(k-1\right) \times s_{ij}.w}{NORM}, \tag{4.3}$$

où $n_i.s(k)$ est l'état du neurone n_i à l'itération k. La constante NORM est un facteur, égal à 9 dans notre cas (le neurone courant et ses 8 voisins), qui normalise les états des neurones à 1. Quand cet algorithme est en fonctionnement, les neurones présents dans les régions les plus texturées (avec un gradient local élevé) de l'image voient leurs variables d'état réduire très vite. À l'inverse, les neurones présents dans des régions homogènes de l'image voient leurs variables d'état se maintenir proche de 1. Après un certain nombre d'itérations, un seuillage est effectué de la façon suivante :

$$n_i.s\left(k\right) \leftarrow \begin{cases} 0 & \text{si } n_i.s(k) \leq THRESH \\ n_i.s\left(k\right) & \text{sinon} \end{cases} \tag{4.4}$$

où THRESH est la valeur du seuil. Dans nos tests, la valeur du seuil a été choisie de façon empirique autour de $\frac{1}{255}$. Nous souhaitions que le seuillage effectué détecte les neurones présents dans les régions de l'image avec beaucoup de contrastes. Nous avons donc ajusté le seuil en testant notre algorithme sur différentes images. Les pixels associés avec les neurones dépassant le seuil ($n_i.s(k) = 0$) sont identifiés comme faisant partie des objets qui seront encodés en sons dans les étapes suivantes du système complet.

4.2 Implémentation du modèle

Dans cette section, nous décrivons la plateforme et les outils de développement. Dans le cadre de ce projet, notre contribution a été d'implémenter en temps réel l'algorithme présenté ci-dessus. Nous avons aussi ajusté différents paramètres (seuil, nombre d'itérations, calcul des poids de connection...) et modifié les sorties de l'algorithme pour que celui-ci fonctionne comme nous le souhaitions.

4.2.1 Choix de la plateforme

Dans le cadre de ce projet, notre objectif était de produire un appareil comprenant une unité de calcul, une caméra et des écouteurs. De plus, il fallait que le système soit portable, accessible et facile d'utilisation. Dans un premier temps, nous avons dû faire face à deux choix possibles :

1. Création d'un système complet en hardware : caméra + unité de calcul + écouteurs.
2. S'appuyer sur un système existant : téléphones intelligents, tablettes et MP3.

Pour des raisons de facilité d'accès, nous avons décidé de choisir la deuxième option et donc de s'appuyer sur un système existant.

Une fois ce choix effectué, nous avions encore un autre choix à faire : quel système d'exploitation utiliser. Les plus connus étant Symbian Nokia™ [Nokia, 2012], RIM BlackBerry™ [BlackBerry, 2012], Android Google™ [Google, 2012], iOS Apple™ [Apple, 2012] et Windows Mobile™ [Microsoft, 2012]. Pour faire notre choix, nous nous sommes appuyés sur une étude des parts de marché datant de 2011 de chacun de ces systèmes d'exploitations afin de savoir lequel était le plus utilisé et donc le plus accessible. D'après le cabinet d'étude IMS Research [IMS-Research, 2012] et Gartner [Gartner, 2012], même si au niveau des téléphones intelligents l'iOS d'Apple™ est à la troisième position avec 18% des parts de marché contre 36% pour Android Google™, au niveau des tablettes tactiles et des MP3, l'iOS d'Apple ™est en première position avec plus de 70% des parts de marché dans ces deux domaines. En outre, d'un côté technique, les appareils d'Apple™ ne sont pas ceux qui utilisent les processeurs les plus puissant mais les outils de développement mis à disposition par Apple sont ceux qui permettent une implémentation plus intuitive.

Aussi, pour respecter nos contraintes d'autonomie, de portabilité et d'accessibilité, nous avons décidé de développer notre système sur iPhone/iPad/iPod™ (iOS Apple™).

4.2.2 Implémentation en Objective-C

À la suite de notre choix pour la plateforme, nous avons dû choisir l'outil de développement. Pour cette étape, un seul choix a été possible car le système d'exploitation des iPhone/iPad/iPod™ (iOS d'Apple™) utilise le langage de programmation orienté objet réflexif Objective-C. C'est une extension du C ANSI, comme le C++, mais qui se distingue de ce dernier par sa distribution dynamique des messages, son typage faible ou fort, son typage dynamique et son chargement dynamique. Contrairement au C++, il ne permet pas l'héritage multiple mais il existe toutefois des moyens de combiner les avantages de C++ et d'Objective-C. Ce langage est basé sur la bibliothèque de classe Cocoa.

Les différentes étapes de notre algorithme sont présentées dans la figure 4.1 et détaillées dans la section qui suit.

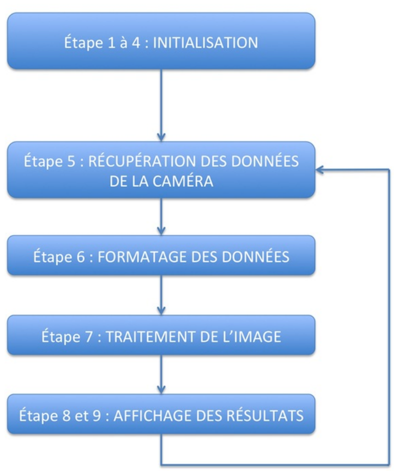


Figure 4.1 Organigramme des différentes étapes de l'algorithme de traitement d'image (mise en valeur des zones importantes).

1. **Création d'une session « AVCaptureSession »** : la première étape de notre développement sous iPhone/iPad/iPod™ consistait à accéder aux images capturées par la caméra. Pour cela, nous avons utilisé les « framework » AV Foundation (voir *AV Foundation Programming Guide*), CoreVideo (voir *CoreVideo Programming Guide*) et CoreData (voir *CoreData Programming Guide*) qui nous permettent tout d'abord de créer une « AVCaptureSession » (code en annexe B.1). Nous avons fait ces choix car AV Foundation permet de capturer facilement de données, de pré-visualiser ces données et se configure très facilement. CoreVideo et CoreData permettent de manipuler les données capturées à l'aide de AV Foundation.

2. **Création d'une vue pour la caméra** : par la suite, nous affichons ce que la caméra enregistre. Cela est pratique car nous pouvons par la suite dessiner la sortie de notre réseau par dessus la vue de la caméra. Pour cela nous créons une vue « PreviewLayer » (figure 4.2 et code en annexe B.2).

Figure 4.2 Exemple d'une vue qui affiche les images enregistrées par la caméra : deux tasses et un Iphone sur une table.

3. **Choix de la caméra** : une fois ces deux étapes effectuées, nous devons définir quelle caméra nous souhaitons utiliser sur l'appareil car sur les iPhone/iPad/iPod™ dernière génération, il y a deux caméras. Dans notre cas, nous sommes intéressés par celle située à l'arrière de l'appareil. Nous indiquons donc quelle caméra utiliser et nous lui assignons une entrée (code en annexe B.3).

4. **Configuration des entrées, des sorties et de la résolution de l'image** : ensuite, nous devons configurer comment capturer la sortie. Nous choisissons notre for-

mat de pixel désiré ainsi que la résolution des images que nous voulons traiter. Dans notre cas, nous utilisons le format défini comme « kCVPixelFormatType32BGRA » (espace des couleurs *Blue Green Red Alpha* encodé en 32 bits). Ce format est celui recommandé par Apple pour capturer des trames vidéos. Pour la résolution de l'image nous utilisons la plus basse possible soit 192x144 pixels (code en annexe B.4) pour que l'algorithme soit le plus rapide possible.

5. **Récupération des données** : nous devons ajouter à la session l'entrée et la sortie que nous avons créées et commencer la simulation (code en annexe B.5). Nous pouvons alors recevoir les images capturées. Pour avoir accès aux données et pouvoir faire notre traitement, nous devons utiliser des fonctions de « CoreVideo » et « CoreMedia ». Ces fonctions nous permettent d'enregistrer les données de l'image depuis la mémoire temporaire (code en annexe B.6).

6. **Formatage des données** : à partir de ce point, nous manipulons les données pour obtenir les pixels en niveau de gris.

7. **Traitement de l'image** : nous effectuons notre traitement d'image comme expliqué en Annexe A.

8. **Création d'une vue superposée à celle de la caméra** : comme à l'étape 2, nous créons maintenant une vue qui affichera la sortie de notre algorithme par dessus l'image capturée par la caméra (figure 4.3).

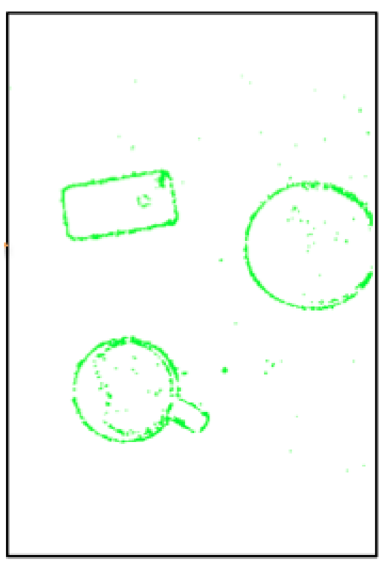


Figure 4.3 Exemple d'une vue qui affiche la sortie de l'algorithme : deux tasses et un iPhone^{TM} sur une table.

9. **Affichage du résultat du traitement d'images** : chaque pixel, associé à un

neurone considéré comme non-actif, est affiché en vert. Cela nous permet de voir directement les parties de l'image mises en valeurs par notre algorithme. Toutes les autres parties de l'image ne sont pas colorées (voir figure 4.4 et Annexe A).

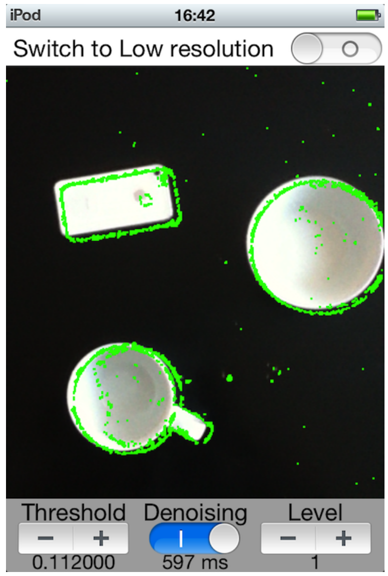


Figure 4.4 Exemple des deux vue superposées : deux tasses et un iPhone™ sur une table.

4.2.3 Problèmes rencontrés

Les outils de développement mis à disposition par Apple nous ont permis d'avoir des résultats très rapidement. Toutefois, plusieurs problèmes sont apparus :

1. Temps de calcul trop long (sur iPhone™ et iPod™) : 250 ms de calcul + 300 ms d'affichage de la sortie de l'algorithme.

2. Sortie de l'algorithme bruitée : mise en valeur des objet peu nette, beaucoup de fausse « détection ».

Le problème du temps de calcul présente deux difficultés : temps de traitement du réseau de neurones et temps d'affichage. La difficulté inhérente au temps de traitement du réseau de neurones vient du fait qu'à chaque parcours de l'image, nous calculons le poids des synapses pour chaque neurone. En ce qui concerne le temps d'affichage, le problème vient du fait que nous affichons trop d'informations à l'écran de l'iPhone/iPad/iPod™ car nous colorons toutes les zones qui n'ont pas d'intérêt.

Le deuxième problème est plus complexe. En effet, pour que l'algorithme soit le plus rapide possible pour notre système temps réel, nous limitons le nombre d'itérations de l'algorithme (une itération entraine un parcours de l'image, soit environ 50ms sur iPhone™ et iPod™). Étant donné que nous limitons le nombre d'itérations, la différenciation des zones est plus complexe. Aussi, dans certains cas un neurone n'a pas le « temps » de se lier à ses voisins assez fortement pour être dans le même état que ceux-ci. Cela provoque ce que l'on appelle « du bruit », c'est-à-dire que la sortie d'un neurone n'est pas représentative étant donné qu'elle diffère de celle de tous ses voisins.

4.2.4 Solutions apportées aux problèmes

Optimisation du temps de calcul

Voici les solutions que nous avons utilisés afin de réduire le temps de calcul :

1. Inversion de l'affichage : il s'agit de colorer uniquement les zones d'intérêt. Pour cela, il nous suffisait simplement d'inverser notre logique au niveau du seuillage. En reprenant l'équation 4.4, l'inverse nous donne :

$$n_i.s\left(k\right) \leftarrow \begin{cases} 0 & \text{si } n_i.s(k) \geq THRESH \\ n_i.s\left(k\right) & \text{sinon} \end{cases} \tag{4.5}$$

2. Pré-enregistrement des poids de connexion des synapses : au lieu de calculer le poids des synapses pour chaque neurone à chaque parcours, ils sont tous pré-calculés durant l'initialisation et stockés dans un tableau. On utilise par la suite la différence entre les pixels comme pointeur dans cette table. Par exemple si $\text{abs}(n_i.p - n_j.p) = 3$ (voir équation 4.2), la valeur 3 définit la valeur du pointeur dans la liste. Dans cette liste, à la case 3, on trouve le résultat de l'équation des poids (Annexe A).

Optimisation de l'affichage

Par la suite, nous avons voulu nettoyer l'image en enlevant le bruit présent dans l'image (ne pas faire apparaitre certains pixels isolés). Pour cela, nous avons décidé d'utiliser un filtre médian [Tukey, 1971] en post-traitement. Son fonctionnement consiste à remplacer la valeur d'un pixel par la valeur médiane de l'ensemble des pixels de son voisinage. Dans notre cas, considérons les 8 pixels voisins ainsi que le pixel, cela nous donne n=9. On ordonne les 9 valeurs de la plus petite à la plus grande. La médiane est alors la valeur placée au milieu de cette suite ordonnée.

4.2.5 Optimisation de l'interface

Pour une meilleure utilisation de l'application développée et afin de permettre un réglage des paramètres plus aisé, nous avons optimisé l'interface de l'application. Toutes ces optimisations sont détaillées dans l'annexe C.

4.2.6 Résultats du traitement d'images

L'optimisation du temps de calcul et de l'affichage discutés précédemment ont permis d'arriver à des résultats très satisfaisants : l'algorithme détecte différentes zones qui semblent importantes dans l'image. Par exemple, dans la figure 4.5 nous pouvons voir que l'algorithme détecte les contours d'objets placés sur un bureau.

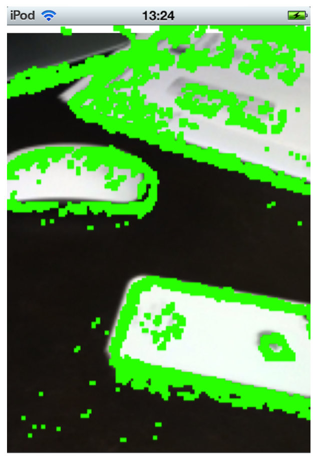


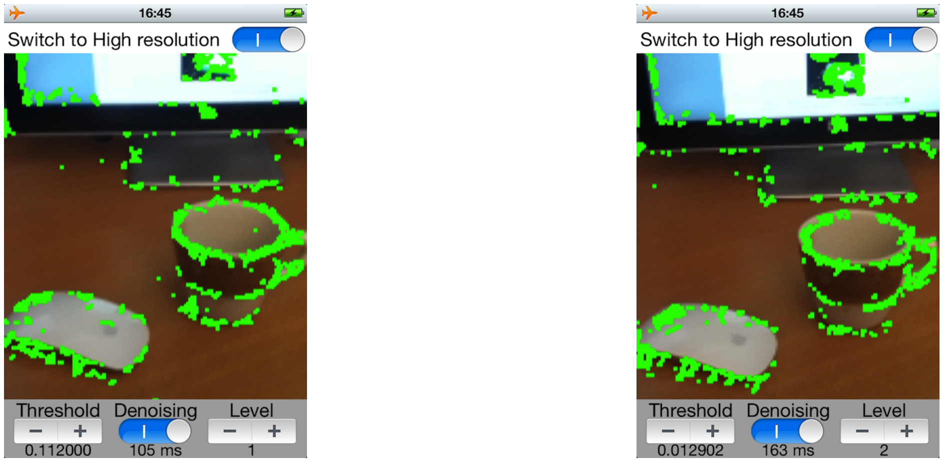
Figure 4.5 Résultat obtenu sur un iPod™ : un iPhone™ et une souris d'ordinateur posés sur un bureau au premier plan, une feuille et le coin d'un écran au second plan.

Le type de résultat obtenu ci-dessus (figure 4.5) est généré très rapidement (tableau 4.1).

Tableau 4.1 Temps de calcul suivant l'appareil utilisé

Appareil	**Temps de calcul sans affichage**	**Temps de calcul avec affichage**
iPod Touch 2/iPhone 4 iOS 5.1	[153 ms ; 162 ms]	[182 ms ; 210 ms]
iPad 2 iOS 5.1	[38ms ; 47ms]	[73 ms ; 80 ms]

Par la suite, l'optimisation de l'interface a permis de mieux paramètrer notre algorithme et ainsi d'ajuster le filtrage et le seuillage pour de meilleurs résultats (figure 4.6). Cette figure nous permet aussi de voir la différence entre deux configurations possibles : l'image (a) a été obtenue après une itération de l'algorithme et l'image (b) après deux itérations. Ce qui est intéressant à voir sur ces deux images est qu'une itération supplémentaire ne permet pas d'obtenir beaucoup plus d'informations sur l'image (avec les paramètres utilisés dans ce cas). Toutefois l'itération supplémentaire ralentit d'environs 60 ms le temps de traitement. Il est à noter que la valeur du seuil joue beaucoup sur la sortie obtenue.



(a) Basse résolution et 1 itération (105 ms) (b) Basse résolution et 2 itération (163 ms)

Figure 4.6 Résultats obtenus sur un iPod™ : une souris d'ordinateur posée sur un bureau au premier plan, une tasse et le coin d'un écran au second plan.

Grâce à ces optimisations, nous avons aussi été en mesure d'adapter et de tester notre algorithme avec une haute résolution (figure 4.7).

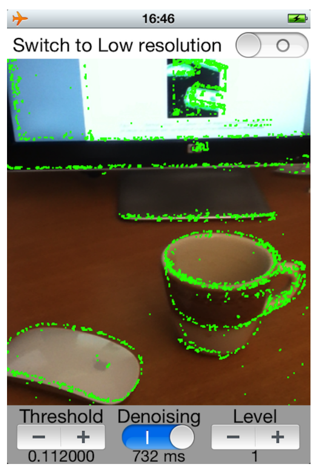


Figure 4.7 Résultat obtenu sur un iPod™ : une souris d'ordinateur posée sur un bureau au premier plan, une tasse et le coin d'un ordinateur au second plan. Ces résultats ont été obtenus en 732 ms avec une haute résolution et avec 1 itération. Les mêmes résultats étaient obtenus en moins de 150 ms avec un iPad™.

4.2.7 Comparaison des résultats

Afin de prouver que notre traitement d'image est valide, nous l'avons comparé à un autre algorithme très connu : détection de Canny ou filtre de Canny [Canny, 1983]. Celui-ci a pour but de trouver les contours dans une image et fonctionne de la façon suivante :

1. Réduction du bruit dans l'image par filtrage gaussien 2D
2. Calcul des gradients d'intensité des contours dans l'image
3. Calcul des orientations des contours
4. Conservation des points correspondants à des maxima locaux
5. Seuillage des contours (seuillage à hysteresis)

Les résultats sont présentés dans la figure 4.8 et comme on peut le voir sur l'écran de l'Ipod, le temps de calcul et d'affichage est de 263 ms soit 158 ms de plus que notre système. L'application développée sous l'iOS d'Apple testée ici a été créée par Robin Summerhill et est disponible sur son site web [Summerhill, 2012]. À noter que celle-ci ne traite pas le flux vidéo.

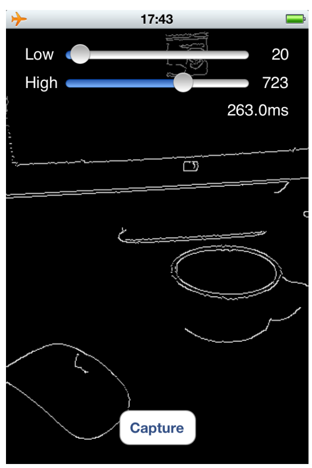


Figure 4.8 Exemple de résultat d'un autre algorithme : détection de Canny ou filtre de Canny.

4.3 Détection des zones actives

Dans cette section, nous détaillons les critères utilisés qui nous permettent d'identifier les zones actives. Les zones actives sont par la suite utilisées dans le système global. En effet, dans le système global, une zone considérée comme active correspondra à un son qui sera localisable comme venant de cette zone. Par exemple, si la zone 4 de la figure 4.9 est active, l'utilisateur entendra un son qui lui semblera venir de la gauche.

4.3.1 Découpage de l'image en zones

Dans un premier temps, nous avons choisi comment découper l'image en zones. Pour cela, nous nous sommes appuyés sur l'étape suivante du système global qui est la génération de sons. Nous avons décidé de découper l'image en neuf zones : trois lignes et trois colonnes (figure 4.9). Cette décision a été prise afin de ne pas trop saturer le système auditif. En effet, plus nous augmentons le nombre de zones dans l'image et plus nous augmentons la quantité d'informations potentielles transmises au système auditif.

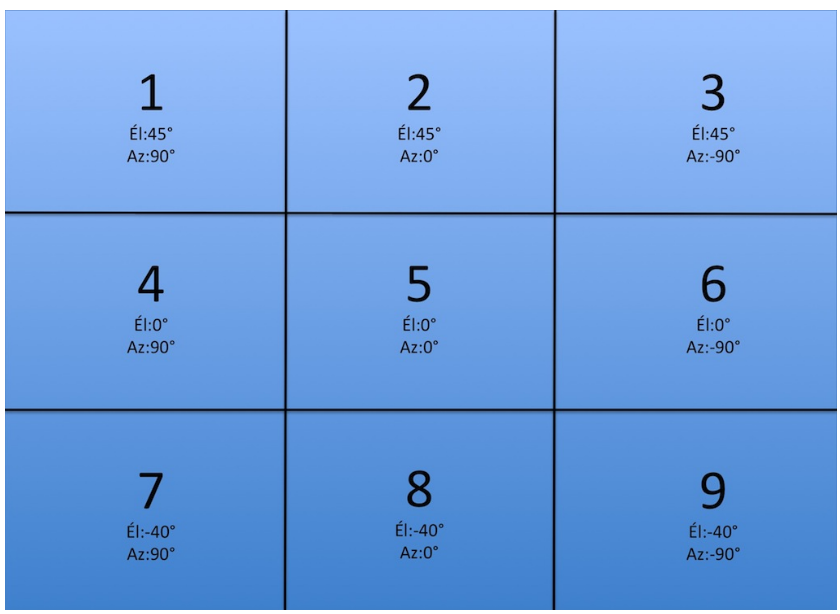
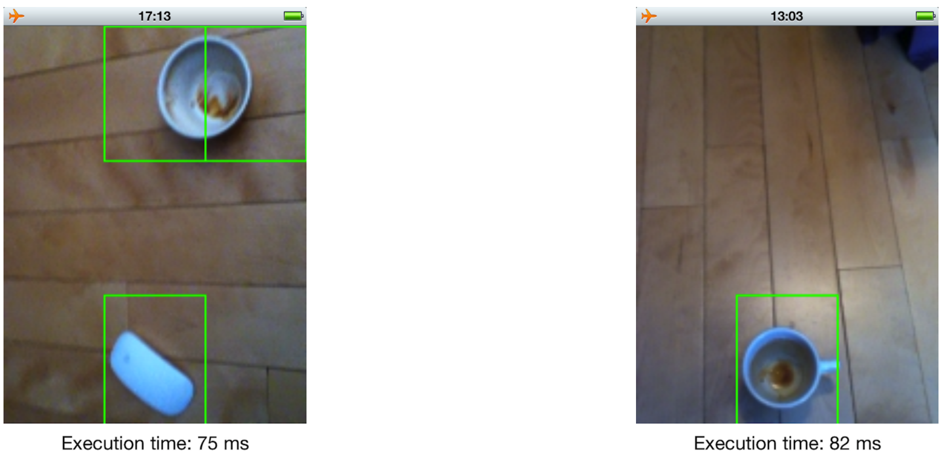


Figure 4.9 Découpage de l'image en différentes zones. Chaque zone correspond à un couple azimut/élévation et est caractérisée par un son filtré. L'élévation maximale est 45°, la minimale est -40°. L'azimut maximal est +90° ou -90°.

4.3.2 Méthode de détection des zones actives

Simplification de l'interface

Afin de détecter les zones actives, nous avons utilisé l'algorithme de mise en valeur des zones importantes détaillé dans l'annexe A. Cependant, l'interface développée (annexe C) n'étant pas nécessaire, nous l'avons simplifiée afin de n'afficher que les zones actives et le temps de calcul à titre d'information (figure 4.10).



(a) Détection d'une tasse et d'une souris (75 ms) (b) Détection d'une tasse (83 ms)

Figure 4.10 Détection et affichage des zones actives.

Critère de détection

Afin de pouvoir détecter et identifier les zones actives, nous avons défini un critère de détection. Celui-ci est détaillé dans l'équation 4.6.

$$taux\_activite_i = \frac{activite\_locale_i}{activite\_globale}, \tag{4.6}$$

Le taux d'activité d'indice i correspond au taux d'activité dans la zone i. L'activité globale correspond au nombre de neurones actif dans toute l'image et l'activité locale i correspond au nombre de neurones actif dans une zone i de l'image. Un neurone est considéré comme actif si son état est nul suite au seuillage effectué à partir de l'équation 4.5 (pixels colorés en vert dans les exemples précédents). Ce critère permet de détecter les zones les plus actives de l'image et ainsi de ne pas considérer certaines parties bruitées comme actives.

L'implémentation de ce critère est effectuée de la façon suivante :

- Dans un premier temps, si un neurone est actif (voir équation 4.5), l'activité globale est incrémentée de un et l'activité locale de la zone dans laquelle il se trouve est elle aussi incrémentée de un (code en annexe D.1).
- Dans un second temps, nous calculons le taux d'activité dans chaque zone d'après l'équation 4.6 (code en annexe D.2).

Seuillage du taux d'activité et affichage des zones actives

Pour finir, la dernière étape consiste à seuiller le taux d'activité de chaque zone afin de déterminer si cette zone doit être considérée comme active ou non (voir équation 4.7).

$$activiteZone_i \leftarrow \begin{cases} TRUE & \text{si } activiteZone_i \geq Seuil \\ FALSE & \text{sinon} \end{cases} \tag{4.7}$$

En d'autres termes, si le taux d'activité à l'intérieur d'une zone est assez importante (par exemple Seuil=15%) alors, cette zone est considérée comme active.

Si une zone est considérée comme active, alors un rectangle vert délimitant cette zone sera affiché (figure 4.10 et code en annexe D.3).

CHAPITRE 5

Génération de sons

5.1 Création du son pour le système global

Dans cette section, nous décrivons les sons que nous utilisons dans le système global et la technique utilisée pour les obtenir.

5.1.1 Description des sons utilisés

Afin de ne pas trop encombrer le système auditif, nous avons limité le nombre de zone à neuf (figure 5.2). Ce nombre de zone pourra être modifié par la suite en fonction des premiers retours des utilisateurs.

Verticalement, l'image est divisée en trois. Aussi, afin de simplifier la localisation sonore verticale qui est la plus compliquée, nous avons décidé d'associer trois sons d'une même octave pour chaque tranche verticale. Chaque zone de la figure 5.2 est caractérisée par un son différent qui est filtré différemment suivant la technique VAS. Nous avons donc généré neuf sons différents. Pour cela, nous avons décidé d'utiliser le logiciel GarageBand d'Apple^{TM} [Apple, 2012] qui permet de générer des notes de musiques de divers instruments (figure 5.1).

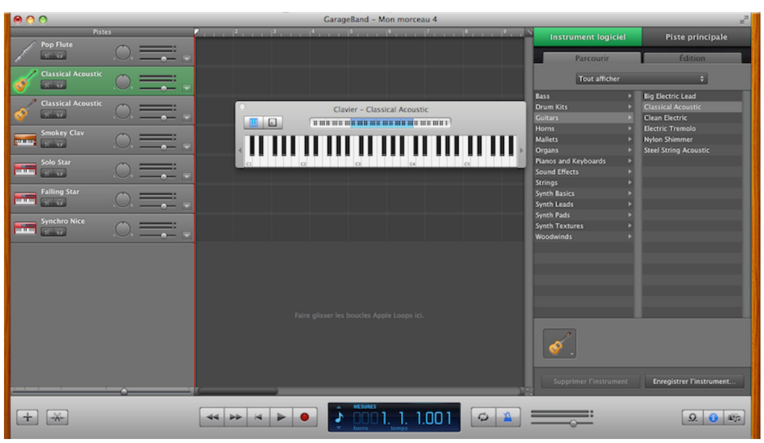


Figure 5.1 Interface du logiciel GarageBand.

Dans le but de tester notre système global nous avons généré neuf notes de flute. Ces trois notes correspondent aux « RÉ », « MI »et « LA » de trois octaves différentes. Les trois sons ont été par la suite enregistrés dans neuf fichiers « .wav ».

5.1.2 Utilisation des sons

Filtrage des sons

Une fois les trois sons enregistrés, nous devons les rendre utilisables dans notre application. Pour cela, ces trois sons doivent être localisables dans l'espace et correspondre à une zone de notre image (figure 4.9). Nous utilisons donc la technique détaillée dans la section 5.2 (technique VAS) afin de rendre ces trois sons localisables. Chaque son est filtré suivant la zone qu'il représente. Nous avons au final neuf sons qui correspondent aux neuf zones de notre image (figure 5.2). Ceci se fait par l'utilisation du logiciel Matlab avec le script détaillé dans l'Annexe E.

Figure 5.2 « Positionnement » des sons filtrés correspondants aux différentes zones : le son son1'.wav correspond au son1.wav filtré par le filtre correspondant à l'azimuth 90° et à l'élévation 45°.

Implémentation de la lecture des sons en Objective-C

Afin de pouvoir utiliser les sons dans notre application, nous avons implémenté leur lecture en Objective-C. Pour cela, à chaque fin de boucle de notre algorithme, si une zone est active, alors un son correspondant à cette zone est joué. Plusieurs étapes étaient nécessaires pour atteindre ce but :

- Initialisation des sons (code en annexe E.7)
- Activation des sons (code en annexe E.8)
- Arrêt des sons (code en annexe E.9) et remise à zéro des sons (code en annexe E.10)

5.2 Étude sur les fonctions de transfert relatives à la tête (*Head Related Transfer Function*)

5.2.1 Choix d'une base de données de fonctions de transfert relatives à la tête

La première étape consistait à valider l'utilisation de fonctions relatives à la tête dans le cadre de ce projet afin de rendre les sons localisables. Pour cela, nous avons utilisé une des bases de données publiques existante, celle de CIPIC [Algazi *et al.*, 2001]. Cette base de données a été mesurée au laboratoire *U. C. Davis CIPIC Interface Laboratory*. Elle donne accès aux réponses impulsionnelles relatives à la tête mesurée pour 90 individus. Ces mesures ont été effectuées pour 25 azimuts différents et 50 élévations différentes, pour un total de 1250 directions différentes.

5.2.2 Résultats obtenus

Les tests effectués à partir de cette base de données nous ont permis d'aiguiller notre choix vers l'utilisation de fonctions relatives à la tête. En effet, après avoir filtré quelques sons avec les bancs de filtres de la base de données, nous avons été en mesure d'écouter et de localiser ces sons. Pour cela, nous (trois personnes au total) écoutions les sons via des écouteurs en essayant de les localiser. Après de nombreux tests, nous en sommes venus à la conclusion que l'utilisation de fonctions relatives à la tête était un bon choix pour le développement de ce projet.

5.3 Création de l'espace acoustique virtuel (VAS, *Virtual Acoustic Space*)

Dans cette section, nous décrivons les méthodes utilisées et les mesures effectuées afin de créer un espace acoustique virtuel.

5.3.1 Justification des mesures de fonctions relatives à la tête

Dans le cadre de ce projet, nous souhaitions utiliser des fonctions relatives à la tête. Pour cela, deux choix se proposaient à nous :

1. Mesurer les fonctions relatives à la tête
2. Utiliser des fonctions existantes relatives à la tête

Les bases de données publiques sont utilisables uniquement pour de la recherche mais ne permettent pas de commercialisation. Afin de ne pas être limité dans le futur, nous avons décidé de mesurer nos propres fonctions relatives à la tête. Ce choix a aussi été fait car nous souhaitions mesurer l'impact de l'utilisation d'écouteurs sur les fonctions relatives à la tête. Plus précisément, nous souhaitions modéliser les écouteurs de Sonomax précédemment présentés. Il faut savoir que les HRTFs sont individuelles vu quelle dépendent de la forme de la tête et de l'oreille externe. Les HRTFs varient largement d'une personne à l'autre et dans notre cas, nous nous sommes limités à mesurer pour une seule physiologie.

5.3.2 Préparation des mesures

Les mesures ont été effectué dans la salle anéchoïque ICAR à l'ÉTS. Dans un premier temps, il a fallu mettre en place la salle anéchoïque (figure 5.3) ainsi que le système d'acquisition de sons. En ce qui concerne la mise en place de la salle anéchoïque, cela consistait à placer des cônes au sol dans toute la salle sauf autour de la tête afin que le bras automatisé puisse se déplacer.

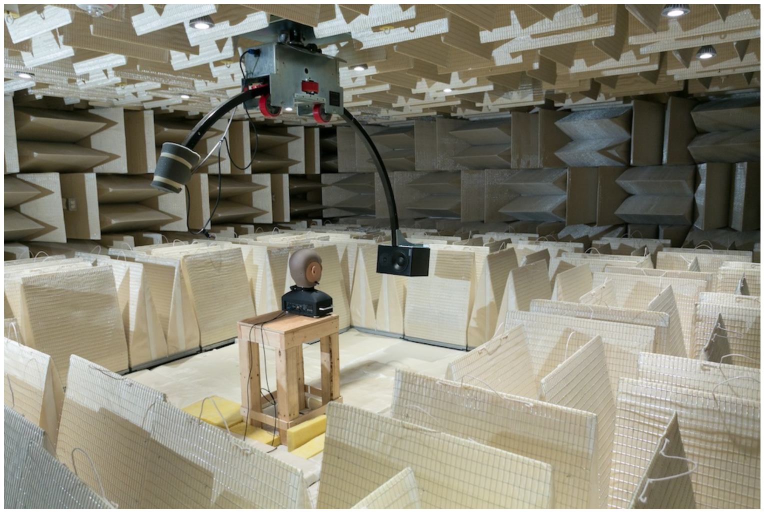


Figure 5.3 Salle anéchoïque ICAR après installation des cônes au sol avec bras automatisé.

La tête artificielle utilisée (figure 5.4) est équipée de deux microphones au niveau des oreilles qui sont accessibles via des connecteurs « Lemo » [G.R.A.S, 2012]. Ces deux connecteurs « Lemo » étaient reliés à un amplificateur avec deux sorties de type « BNC ». L'acquisition des deux microphones de la tête se faisait à partir de deux câbles de type « BNC » reliés à un ordinateur équipé d'une carte d'acquisition (figure 5.5).

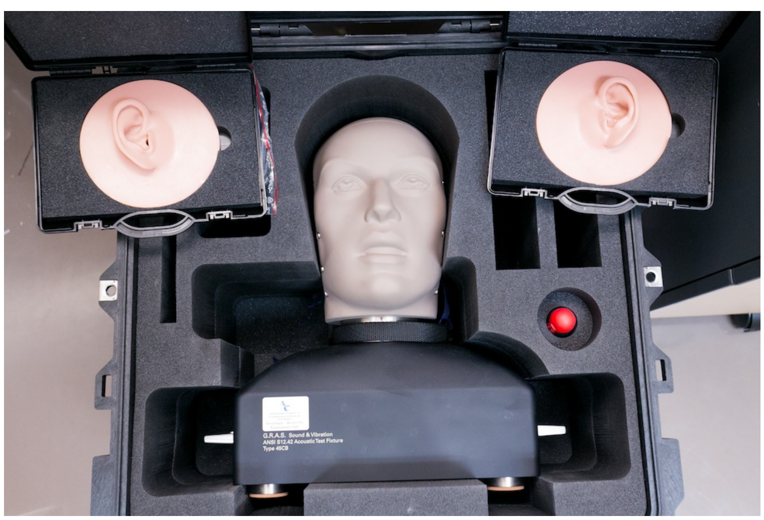


Figure 5.4 Tête G.R.A.S 45CB utilisée afin de mesurer les HRTFs.

Nous avons créé un fichier texte de format .txt (figure 5.6) contenant les positions auxquelles devaient se déplacer le bras automatisé (développé par Daniel Audio Labs) [Daniel-Audio-Labs, 2012]. Ce fichier devait être chargé dans une application LabVIEW$^{\text{TM}}$ [National-

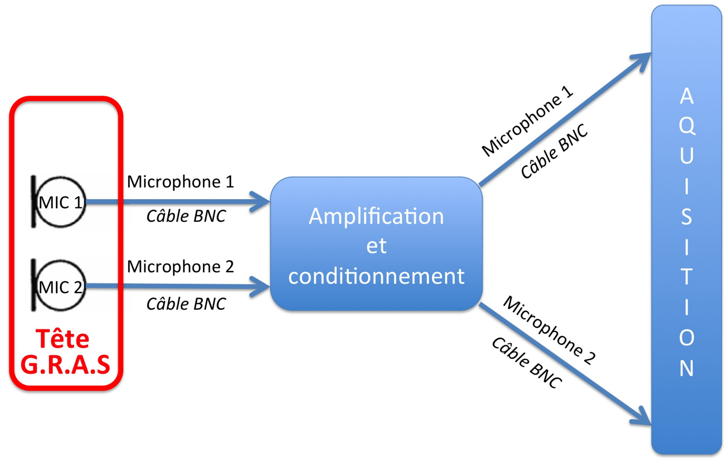


Figure 5.5 Schéma bloc du système d'acquisition et de ses connectivités.

Instruments, 2012] qui gérait le déplacement du bras robotisé. Nous avons créé et testé un fichier texte qui permettait au bras automatisé de couvrir de 0° à 180° horizontalement et de -40° à 45° verticalement par pas de 45°.

1.0000000e+00	0.0000000e+00	4.5000000e+01	1.5000000e+00
2.0000000e+00	0.0000000e+00	-4.0000000e+01	1.5000000e+00
3.0000000e+00	9.0000000e+01	-4.0000000e+01	1.5000000e+00
4.0000000e+00	9.0000000e+01	0.0000000e+01	1.5000000e+00
5.0000000e+00	9.0000000e+01	4.5000000e+01	1.5000000e+00
6.0000000e+00	2.7000000e+02	4.5000000e+01	1.5000000e+00
7.0000000e+00	2.7000000e+02	0.0000000e+01	1.5000000e+00
8.0000000e+00	2.7000000e+02	-4.0000000e+01	1.5000000e+00

Figure 5.6 Exemple de fichier texte (.txt) pour le déplacement du bras automatisé. La première colonne contient le numéro du déplacement, la deuxième la position horizontale (en degré), la troisième la position verticale (en degré) et la dernière le temps d'attente entre chaque déplacement (non utilisé).

Une fois ces préparations terminées, nous étions prêt à effectuer les mesures que nous détaillons dans les sections suivantes.

5.3.3 Acquisition des données

La procédure complète d'acquisition des données est précisée dans la figure 5.7.

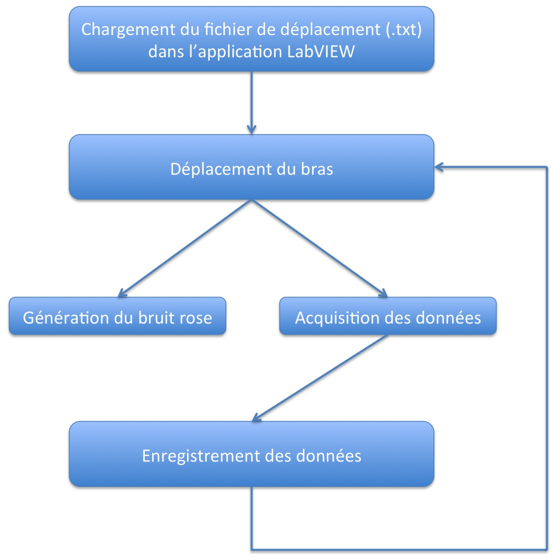


Figure 5.7 Schéma bloc de la procédure d'acquisition des données.

Afin de mesurer les fonctions relatives à la tête, nous devions générer un son identique à différentes positions et enregistrer ce son filtré par les oreilles du mannequin. Pour cela, nous avons généré un bruit rose (figure 5.8 et code en annexe E.1) qui était ensuite joué à différents endroits du champ visuel de la tête G.R.A.S. Nous avons fait ce choix car le bruit rose est un signal aléatoire dont l'intensité de chaque bande de fréquence de taille identique est égale, si on l'analyse dans une échelle logarithmique. Autrement dit, il possède une intensité constante par octave. La perception humaine étant logarithmique, le concept de bruit rose est plus adapté à la perception humaine que celui de bruit blanc.

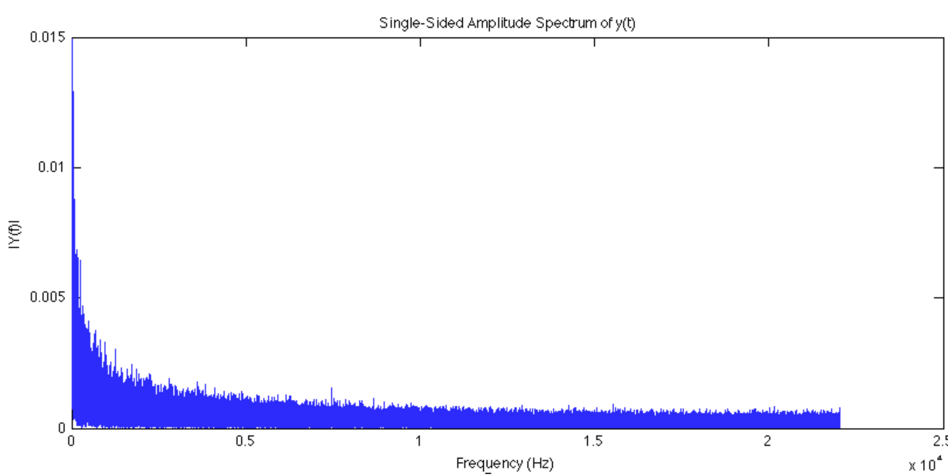


Figure 5.8 Spectre d'amplitude du bruit rose généré.

Une fois le bruit rose généré, nous devions configurer les entrées et les sorties du système d'acquisition (code en annexe E.2). Nous avions une sortie à configurer qui permettait de faire jouer le bruit rose sur le haut-parleur placé au bout du bras automatisé. Puis nous avions soit deux entrées à configurer (correspondantes aux microphones placés sur la tête G.R.A.S), soit une seule entrée à configurer (correspondante au microphone seul placé au milieu de la salle) pour mesurer le signal de référence.

Une fois toutes ces étapes terminées, l'acquisition des données (code en annexe E.3) était possible. La dernière étape consistait à enregistrer les signaux obtenus sous forme de fichier de données Matlab.

Suite à l'acquisition des données, celles-ci étaient enregistrées à partir de la variable *Input_ data* (code en annexe E.3).

5.3.4 Mesures effectuées

Afin de pouvoir calculer les fonctions relatives à la tête, nous avons effectué deux mesures différentes, ainsi qu'une troisième afin de modéliser les bouchons ARP.

Mesure de la référence

Dans un premier temps, nous avons dû mesurer le signal de référence. Pour cela, afin de ne pas modéliser le haut-parleur, nous avons placé un microphone seul au même niveau que la tête G.R.A.S (figure 5.9).

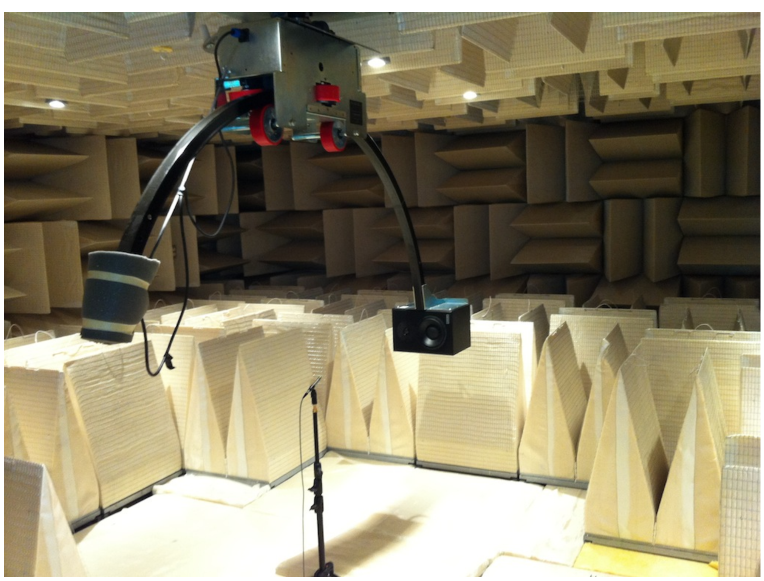


Figure 5.9 Mesure des signaux de références avec microphone au centre.

Nous avons mesuré un signal de référence pour chaque couple azimuts/élévation.

Mesure des signaux à l'intérieur des oreilles

Nous avons mesuré les signaux filtrés par les oreilles. Pour cela, nous avons utilisé la tête G.R.A.S équipée de deux oreilles « standard » (figure 5.10).

Figure 5.10 Mesure des signaux à l'intérieur des oreilles du mannequin.

Nous avons mesuré deux signaux à l'intérieur des oreilles (un signal par oreille) pour chaque couple azimuts/élévation.

Mesure du signal à l'intérieur d'une oreille bouchée par un écouteur Sonomax

Afin de modéliser la fonction de transfert des écouteurs ARP qui vont être utilisés, nous avons répété les mêmes mesures que précédemment avec cette fois-ci un écouteur ARP placé dans une oreille (figure 5.11). La présence d'un microphone à l'extérieur des écouteurs nous a permis de rendre les écouteurs transparents. Nous pouvions ainsi mesurer les sons à l'aide des microphones de la tête artificielle et les comparés avec les sons mesurés sans les écouteurs.

Ces mesures nous permettent de connaitre l'effet des écouteurs et de le corriger dans le cas où nous souhaitons reproduire l'environnement sonore extérieur. De la sorte, l'environnement sonore extérieur n'est pas déformé et la personne équipée de ce système pourra avoir une idée bien précise de ce qui se passe autour d'elle.

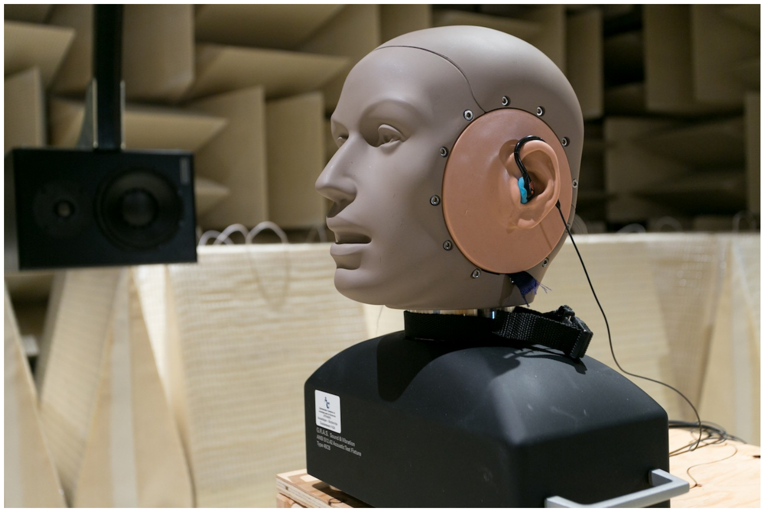


Figure 5.11 Mesure du signal à l'intérieur d'une oreille bouchée par un écouteur ARP.

5.3.5 Traitement des résultats

Afin de traiter les résultats, nous avons utilisé Matlab™ [Mathworks, 2012]. Le traitement des mesures effectuées avait pour but de calculer la réponse inpulsionnelle du filtre réalisé par chaque oreille. Dans un second temps, il s'agissait de faire un produit de convolution entre cette réponse impulsionnelle et un son simple afin d'externaliser ce dernier. Par externaliser, nous voulons dire que le son semble venir de l'endroit désiré. Suite à cette manipulation, en écoutant le son convolué avec la réponse impulsionnnelle, nous (trois personnes) avons été en mesure de localiser différentes sources sonores. Cette validation a été faite de façon subjective par l'écoute de différents sons filtrés à partir de nos mesures.

Calcul des réponses impulsionnnelles

Nous devions tout d'abord charger les données des mesures effectuées (référence et à l'intérieur de l'oreille) puis différencier l'oreille gauche de l'oreille droite (code en annexe E.4). Une fois ces étapes effectuées, nous chargions un son (ici un beep) qui allait nous servir à tester notre filtre.

La suite consistait à estimer la fonction de transfert entre le signal de référence et le signal mesuré à l'intérieur de l'oreille puis à calculer la réponse impulsionnelle inverse du filtre estimé (code en annexe E.5). Soit les signaux acoustiques P_g mesurés dans le conduit auditif gauche, P_d mesurés dans le conduit auditif droit, et P_i mesurés à l'emplacement du centre de la tête, retirée du champ : les fonctions de transfert relatives à la tête sont $\frac{P_g}{P_i}$ à gauche et $\frac{P_d}{P_i}$ à droite. Ces fonctions de transfert relatives à la tête calculées, il suffit d'inverser cette fonction de transfert (figure 5.12) et d'en calculer sa réponse impulsionnelle. Une fois la réponse impulsionnelle calculée, nous filtrions le son pour chaque oreille grâce à la fonction *filter* de Matlab^{TM}.

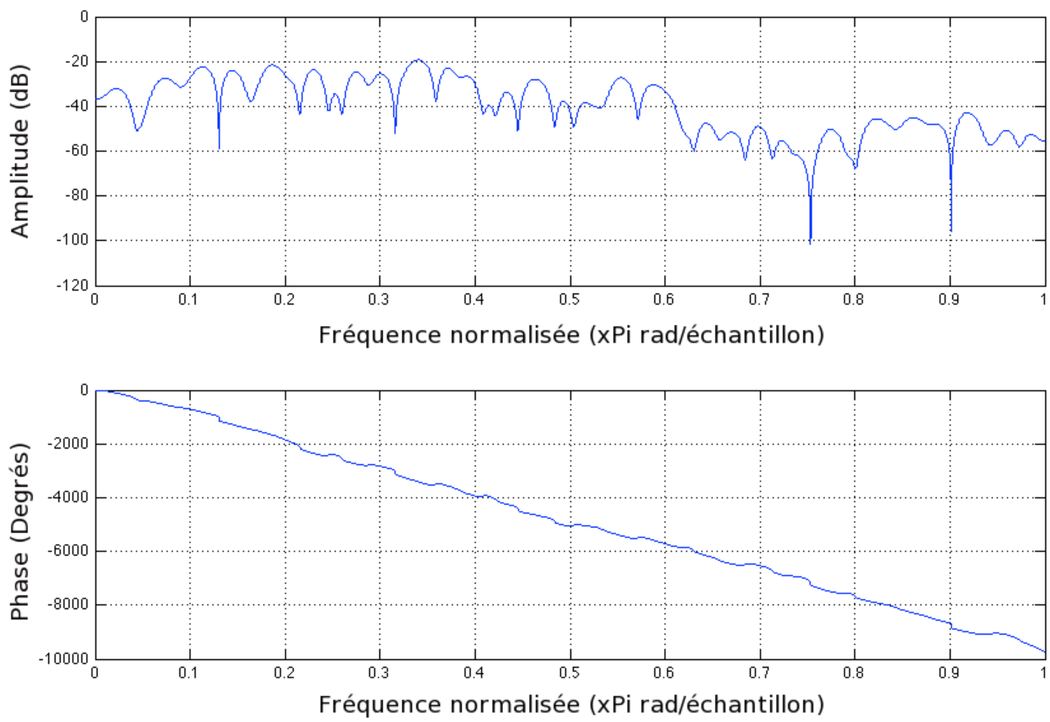


Figure 5.12 Exemple d'une réponse fréquentielle de fonction de transfert inverse. Cette fonction de transfert a été calculée pour une élévation de 45° et un azimut de 0° à partir de mesures sur l'oreille droite du mannequin.

Création du son

La dernière partie consistait à générer un son stéréophonique filtré. Nous enregistrions ensuite ce son dans un fichier afin de pouvoir l'écouter (code en annexe E.6).

Nous avons ensuite répété cette étape pour tous les couples azimuts/élévation afin d'obtenir le champ visuel au complet par tranche de 45°. Nous avons donc obtenu un son filtré par zone (figure 5.13).

Une fois toutes ces étapes achevées, le « Virtual Acoustic Space » était créé.

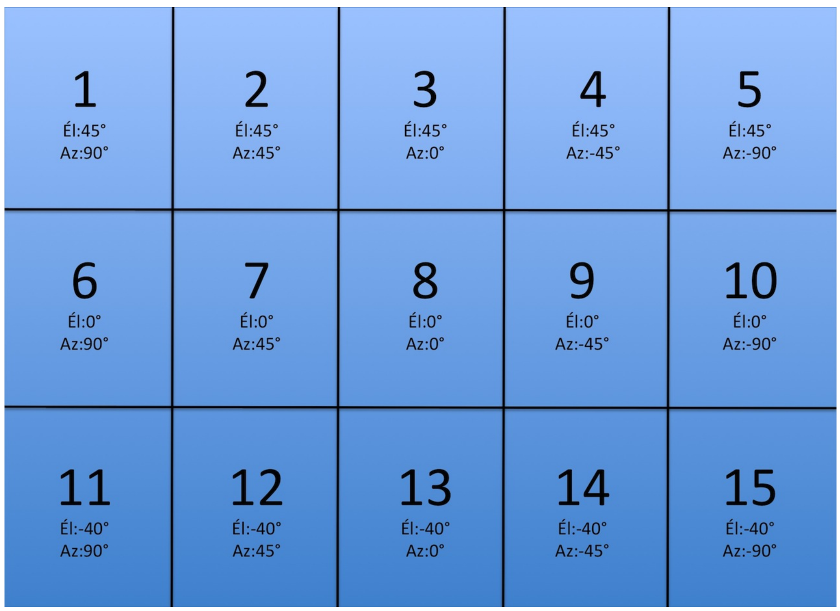


Figure 5.13 Découpage du champ visuel en différentes zones. Chaque zone correspond à un couple azimut/élévation et est caractérisée par un filtre. L'élévation maximale est 45°, la minimale est -40°. L'azimut maximal est +90° ou -90°.

CHAPITRE 6

Test du système

Une population de cinq personnes volontaires a testé et de validé le système dans le cas d'un déplacement dans un milieu encombré puis dans le cas de localisation d'objets placés sur une table. Les personnes ne présentaient pas de déficiences visuelles mais avaient les yeux bandés tout au long des expériences. Afin de valider notre système de génération de sons, une dernière expérience a été menée. Au cours de cette dernière expérience, nous avons demandé aux personnes de localiser verticalement et horizontalement un son filtré à partir de nos mesures.

6.1 Première expérience : déplacement dans un milieu encombré

6.1.1 Présentation de l'expérience

Dans cette première expérience, les personnes devaient partir d'une pièce et accéder à une autre pièce, guidées uniquement par les sons émis par notre système. De plus des obstacles (trois objets au sol) étaient placés dans le corridor séparant les deux pièces. La personne avait cinq essais pour se rendre d'une pièce à l'autre et les objets étaient déplacés à chaque tentative. Les sujets connaissaient préalablement le parcours mais ne connaissaient pas la localisation des obstacles.

6.1.2 Déroulement de l'expérience

L'expérience s'est déroulée en suivant ces étapes :

- Présentation du fonctionnement du système
- Explication du déroulement de l'expérience
- Bandage des yeux
- Début de l'expérience

6.1.3 Résultat de l'expérience

Les résultats sont présentés dans le tableau 6.1. Les commentaires des participants ont été très intéressants : les personnes trouvaient qu'il était facile de localiser les sons mais que se laisser guider par ceux-ci était plus compliqué. Ils avaient tendance à placer la main ne tenant pas le iPod devant eux pour se protéger et palper qu'il n'y ait pas de mur devant eux.

Tableau 6.1 Résultats de la première expérience : nombre d'objets heurtés durant l'expérience. Au total, trois objets étaient disposés dans le corridor.

	Essai 1	**Essai 2**	**Essai 3**	**Essai 4**	**Essai 5**
Sujet 1	2	1	**0**	1	**0**
Sujet 2	1	**0**	**0**	**0**	**0**
Sujet 3	2	1	1	**0**	1
Sujet 4	3	1	2	1	1
Sujet 5	**0**	1	2	1	**0**

6.1.4 Discussion

En considérant les cinq essais de chaque sujet, le taux de réussite est de 36% sans heurter d'objets et de 80% en heurtant au plus un objet. Cependant, en regardant les résultats pour les deux derniers essais de chaque sujet, nous pouvons voir que le nombre d'objets heurtés est au maximum de 1. Cela nous laisse donc penser que ce système requiert de l'apprentissage.

6.2 Deuxième expérience : localisation d'objets sur une table

6.2.1 Présentation de l'expérience

Dans cette seconde expérience, les personnes étaient assises devant une table et devaient localiser trois objets différents. La personne avait 30 secondes pour localiser le maximum d'objets. La personne avait cinq essais pour localiser les trois objets sachant qu'à chaque fois les objets étaient déplacés.

6.2.2 Déroulement de l'expérience

L'expérience s'est déroulée en suivant les mêmes étapes que lors de la première expérience.

6.2.3 Résultat de l'expérience

Les résultats sont présentés dans le tableau 6.2. Comme le montrent ces résultats, la tâche demandée dans cette expérience s'est avéré très facile. Aussi pour ajouter de la difficulté à l'expérience, lors de l'essai n°5 deux des objets à localiser ont été rapprochés. Deux des cinq participants ont réussi à les distinguer en approchant l'Itouch plus près des objets.

Tableau 6.2 Résultats de la première expérience : nombre d'objets localisés. Au total, trois objets étaient disposés sur la table.

	Essai 1	**Essai 2**	**Essai 3**	**Essai 4**	**Essai 5**
Sujet 1	**3**	**3**	**3**	**3**	2
Sujet 2	**3**	**3**	**3**	**3**	**3**
Sujet 3	**3**	**3**	**3**	**3**	2
Sujet 4	**3**	**3**	**3**	**3**	**3**
Sujet 5	**3**	**3**	**3**	**3**	2

6.2.4 Discussion

Avec 88% de réussite pour cette expérience, nous pouvons conclure que le système est performant pour localiser des objets sur une surface plane. À noter que lors du cinquième essai, deux objets étaient placés à une très proche distance l'un de l'autre. Le rapprochement de l'iPod vers la table a permis à deux des sujets de différencier les deux objets.

6.3 Troisième expérience : localisation de sons

6.3.1 Présentation de l'expérience

Dans cette troisième expérience, les participants étaient assis et devaient localiser des sons. Le participant devait donner une localisation verticale (haut, milieu ou bas) et une localisation horizontale (gauche, milieu ou droit).

6.3.2 Déroulement de l'expérience

L'expérience s'est déroulée en suivant ces étapes :

- Explication du déroulement de l'expérience
- Début de l'expérience
- Retour de l'utilisateur

- Reprise de l'expérience avec des azimuts et des élévations différentes

6.3.3 Résultat de l'expérience

Les résultats sont présentés dans le tableau 6.3. Les trois premiers sons étaient présentés dans le plan horizontal (droite, gauche ou centre) et les deux derniers n'importe où dans l'espace. On voit donc que les sons sont plus facilement localisables sur le plan horizontal que sur le plan vertical.

Tableau 6.3 Résultats de la première expérience : OUI pour localisation correcte et NON pour localisation incorrecte. A représente l'Azimut (-90°<=>gauche, 0°<=>centre et 90°<=>droite) et É représente l'élévation (-40°<=>en bas et 45°<=>en haut)

	Essai 1 **A=-90° É=0°**	**Essai 2** **A=90° É=0°**	**Essai 3** **A=0° É=0°**	**Essai 4** **A=0° É=-40°**	**Essai 5** **A=-90° É=45°**
Sujet 1	**OUI**	**OUI**	**OUI**	NON	NON
Sujet 2	**OUI**	NON	**OUI**	**OUI**	NON
Sujet 3	**OUI**	**OUI**	NON	NON	**OUI**
Sujet 4	NON	**OUI**	**OUI**	**OUI**	**OUI**
Sujet 5	**OUI**	**OUI**	NON	NON	NON

6.3.4 Discussion

Avec 73% de bonnes localisations dans le plan horizontal et seulement 40% de bonnes localisations dans le plan vertical, nous pouvons conclure que la localisation sur le plan vertical est la plus complexe. De plus, nous pouvons aussi constater que considérant le fait que nos HRTFs ne sont pas individualisées, elles semblent être plus adaptées au sujet 4 qu'au sujet 5.

CHAPITRE 7
CONCLUSION

7.1 Sommaire

Ce projet vient s'inscrire dans la lignée des systèmes de substitution sensorielle de la vision vers l'audition. À la différence des autres systèmes, celui-ci s'approche le plus possible du système de traitement de l'oeil humain. En effet, nous essayons avec notre algorithme de prendre en considération uniquement certaines zones de l'image qui semblent importantes avec une charge minimale sur le processeur. En ce qui concerne la génération de son, elle se fait en fonction de caractéristiques bien particulières liées au système auditif humain. Ainsi, les sons que nous générons sont agréables à l'oreille et facilement localisables par le système auditif humain.

De plus, l'utilisation de réseau de neurones artificiels qui est une technologie jeune permet de poursuivre et de mettre en valeur les différents travaux développés au sein du laboratoire NECOTIS. Ce projet aura aussi permis d'ouvrir des portes pour des collaborations avec d'autres laboratoires de recherche ne travaillant pas forcément dans le même domaine.

7.2 Contribution

Le développement de ce projet a permis d'implémenter un modèle de réseau de neurones sur un système embarqué (iPhone/iPad/iPod^{TM}). Nous avons prouvé qu'une telle implémentation était possible tout en respectant la contrainte de temps réel. En effet, un système temps réel complet a été développé : celui-ci comprend un iPhone/iPad/iPod^{TM}, un boitier de gestion du système (volume et marche/arrêt du système) et des écouteurs « évolués » (écouteurs ARP de Sonomax).

Au regard des travaux déjà réalisés, notre système de substitution sensorielle de la vision vers l'audition se distingue à différents niveaux :

- **Traitement d'image** : notre système se veut être un modèle du système visuel adapté pour des contraintes de rapidité de traitement. Nous cherchons à réduire la quantité d'information transmise aux système auditif tout en augmentant sa qualité. En effet, notre système ne transmet des informations sonores que sur certaines parties de l'image qui sont considérées comme importantes et non sur toute l'image.

- **Génération de sons** : notre système permet aux utilisateurs de pouvoir localiser facilement nos sons verticalement et horizontalement. Ceci est permis grâce à l'utilisation de la technique VAS.

- **Prototype I** : il permet aux utilisateurs de disposer d'une plateforme facile d'accès : iPod Touch, iPhone ou iPad, équipés d'écouteurs conventionnels.
- **Prototype II** : celui-ci est équipé d'écouteurs ARP de sonomax qui permettent aux utilisateurs de ne pas être coupés de l'environnement sonore vu que celui-ci est reproduit grâce aux microphones présents sur la partie extérieure des écouteurs.

7.3 Travaux futurs

Dans la continuité de ce projet, les prochains travaux à réaliser sont :

- **Test du système** : afin d'identifier les points principaux à améliorer, il faudrait dans un premier temps tester notre système sur une population d'individus ayant des déficiences visuelles afin de recueillir leurs impressions et modifier notre système en fonction de celles-ci.

- **Traitement d'image** : au niveau du traitement d'image, suivant les premiers retours des utilisateurs, il faudrait ajuster les différents paramètres de l'algorithme afin de donner plus ou moins d'information.

- **Génération de sons** : au niveau de la génération de sons, plusieurs points peuvent être améliorés comme le nombre de sons/zones, le type de sons ou encore la qualité des sons. Pour cela, peut-être serait-il bon de faire appel à un compositeur afin de générer des sons plus complexes et plus agréables.

- **Prototype** : lors de la réalisation du prototype, nous avons été confrontés à des problèmes d'accès aux circuits internes du connecteur dock pour iPhone/iPad/iPod™.

En effet, l'accès à ces documents nous aurait permis d'interfacer directement les écouteurs ARP de sonomax avec l'iPod^{TM} sans avoir à passer par un boitier pour gérer les deux microphones. Ce point là pourrait donc être revu en signant des ententes de partenariat avec la compagnie Apple.

- **Plateforme de développement** : il serait également intéressant de développer cette application sur une autre plateforme telle qu'Android afin de comparer les résultats.

ANNEXE A

Article ISSPA, *The 11th International Conference on Information Sciences, Signal Processing and their Applications*, 2-5 Juillet 2012, Montréal, Quebec, Canada

NEURAL VISUAL OBJECTS ENHANCEMENT FOR SENSORIAL SUBSTITUTION FROM VISION TO AUDITION

Damien Lescal, Louis-Charles Caron and Jean Rouat

NECOTIS, Université de Sherbrooke
GEGI, Sherbrooke QC, Canada, J1K 2R1
http://www.gel.usherbrooke.ca/necotis
damien.lescal, louis-charles.caron, jean.rouat[@usherbrooke.ca]

ABSTRACT

The purpose of this research is to develop a sensorial substitution system from vision to audition. The intention is to provide a noninvasive solution for people with visual impairment to compensate their disability using brain plasticity. The "prosthesis" is a signal processing system that analyses the visual scene to identify objects of interest and encode them into sound. The encoding should be based on characteristics of the human auditory system so that the generated sounds provide an overview of the visual scene in front of the patient, enabling him to locate each identified object. This should allow people with visual disabilities to move around more easily, even in cluttered environments. This paper describes the image processing to enhance objects from images and gives an overview of the sound encoding.

1. INTRODUCTION

1.1. Sensorial substitution system

Visual and auditory prostheses involve surgeries that are complex, expensive and invasive. They are limited to a small number of electrodes and can only be used when the impairment is peripheral. Noninvasive prostheses (sensorial substitution systems) have existed for more than 40 years but have not been well accepted in the disability sector. Several systems have been developed since the emergence of this concept. Paul Bach-Y-Rita proposed a substitution system from vision to touch [1] in which pictures captured by a camera were converted into electrical stimulation of the tongue. Other studies have shown that simple tasks such as object position [2], shape recognition [3, 4] and reading [5] can be achieved using vision-to-touch substitution devices.

More recently, substitution systems from vision to audition have been proposed: [6, 7, 8]. The most important systems developed so far are the vOICe [9], PSVA (Prosthesis for Substitution of Vision by Audition) [10], the device developed by Cronly-Dillon [11] and the Vibe [8]. These systems encode a full image with no prior analysis of the visual scene. Usually these systems encode the luminosity of all pixels from the image in the amplitude of modulated sounds. The vOICe and the Cronly-Dillon device use left-to-right scanning to encode horizontal position with interaural time difference (ITD). The Vibe and PSVA encode the entire image in one complex sound. The PSVA uses frequencies that are associated with each pixel and increase from left to right and from bottom to top of the image. The Vibe splits the image into several parts that are equivalent to "receptive fields". Each "receptive field" is associated with a single sound and the sum of all sounds forms a complex sound transmitted to the two ears. The "receptive fields" design is inspired by work on the retina [8].

The challenge in this project resides in the design of a suitable encoding of the visual scene into auditory stimuli such that the content of the sound carries the most important characteristics of the visual scene with no encoding of the full image. These sounds should be shaped in a way that the subject can build mental representations of visual scenes even if the information carrier is the auditory pathway. A sensorial substitution system using an object-based approach is described. An image segmentation algorithm using a neural network combined with sound generation is proposed. That neural system has a good potential for object based image and visual scene analysis. Also, human auditory features such as interaural time difference (ITD) and interaural level difference (ILD) are used to synthesize sounds that better characterize the visual scene for real-life environments. For this purpose, we are using the virtual acoustic space (VAS) [12], also known as virtual auditory space. It is a technique in which sounds presented over headphones appear to originate from any desired direction in space. The illusion of a virtual sound source outside the listener's head is created.

1.2. Object based approach

The automatic location and search of objects in images is still an open issue and is a very active area in image processing research. Recent methods like the ones proposed in [13] find and track objects from images and video by using prior knowledge of the object's structure. Other Bayesian statistical models represent structure of objects as graphs [14]. However, they need to train on large sets of data and require prior knowledge of the visual environment. Objects recognition and location have been de-

978-1-4673-0382-8/12/$31.00 ©2012 IEEE

veloped based on models of vision like HMAX [15] and hierarchical systems [15, 16]. Some of the current methods are described in [17] and softwares like openCV [18] are also widely used. Image segmentation and objects location based on models of binding [19, 20] can now be performed in real time with networks of spiking neurons on embedded systems [21]. Even if this approach is becoming mature, we are looking here for a faster analysis of the images to free CPU processing time for the sound generation of the substitution system. To this end, we propose to preprocess images such that objects or "saliency objects" are enhanced. Then the object retrieval and location should be easier. After that enhancement, the dynamic link matching system [22] could be used to more precisely locate objects in images.

We define saliency regions as areas rich in contrasts or dense in contours and we assume here that objects of interests have such characteristics. We describe in this work the enhancement algorithm and give a global view of the proposed substitution system.

2. DESCRIPTION OF THE SUBSTITUTION SYSTEM

2.1. Overview of the substitution system

In this section, we describe the global system from figure 1.

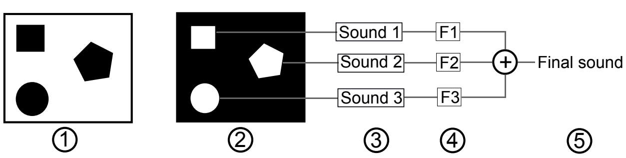


Fig. 1. Block diagram of the system. Modules 1 and 2 for image preprocessing, modules 3–5 for sound generation

1. **Captured image**: each pixel of the image is captured and then represented by one neuron. The closer the gray level between two pixels is, the stronger the connection between the two corresponding neurons is.

2. **Enhanced saliencies**: strongly connected neurons form clusters which represent regions of interest in the image.

3. **Sound generation** : each region is then associated to a simple sound with the following form:

$$S_j = A_j.sin(\omega.t + \phi) \tag{1}$$

where :

- S_j is the sound of one region
- A_j depends on the size of the region
- ω depends on the average level of gray of the region
- ϕ is fixed between 0 and $\frac{\pi}{2}$

4. **Filtering**: every single sound is then filtered depending on its position in the image using Virtual Acoustic Space (VAS) model [12].

5. **Complex sound**: all simple sounds are added and form a complex sound which represents the visual scene

For now the distance between the listener and the object is not encoded. As a prototype, we assume here that the object's size is sufficient. Distance integration is left as a future work.

2.2. Object enhancement algorithm

For object enhancement, a neural network has been developed. Each pixel of the input image is a neuron in the network and synapses connect each neuron to its 8 neighboring neurons, as shown in figure 2.

2.2.1. Neurophysiological model

The model presented in following subsections has been inspired from our knowledge in neurophysiology. In some way, it is equivalent to a recurrent spiking neural map for feature extraction – but much faster. The neurophysiological spiking neural network would comprise *fast local excitatory* and *slower global inhibitory* synapses. Neurons firing first (those who reach first a threshold $THRES$) would contribute immediately to each neuron it is connected. Contributions are important between two neurons that are strongly connected. Inhibition would then take place and reduce transmenbrane electrical potential of all neurons – except for those that are in synchrony with the firing neuron. The excitatory contribution received by a neuron i and emitted by neuron j would be $\delta_j.w_{ij}$. With δ_j, the action potential emitted by neuron j and w_{ij} the connection strength between neurons i and j. The global inhibition factor would be equal to $THRES$ in the spiking neural network implementation.

2.2.2. Implemented model

For a fast implementation and quick execution time, the spiking neural network has not been implemented, but approximated with neurons which outputs are continuous time variables. Non-linear thresholding of the neural network has been preserved in the implemented model. In this model, a neuron has a state variable and a pixel value. A synapse is characterized by a weight which depends on the difference in the pixel value of the two neurons it connects. An algorithm iteratively updates the state of the neurons depending on their current state, the weight of their synapses and the state of their neighbors. The goal of this algorithm is to differentiate homogeneous regions of the image from the ones with dense contours. These regions are considered as objects which will be encoded into sounds in the subsequent steps of the sensorial substitution system. In some way, the algorithm performs a spatial integration of local gradients with no need of explicit edge detections. In principle, this approach is more robust to contrast changes. It is also possible to automatically adapt the threshold depending on the contrast in the image. The following subsections detail how the network is created, describe the iterative algorithm and demonstrate its use for object enhancement with different images.

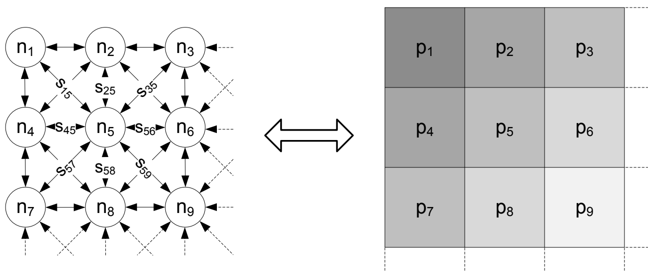


Fig. 2. Representation of an input image as a neural network for object enhancement. Each pixel p of the image has a neuron n associated with it. Neurons are connected to their 8 neighbors by synapses s. Only neuron n_5's synapses are identified in the figure.

2.2.3. *Initialization of connecting weights*

Each pixel of an input image is represented by a neuron in the network. Synapses connect each neuron to its 8 neighboring neurons. The state of the neurons is initialized to 1. The weight of the synapse connecting two neurons is computed based on the difference of the neurons' pixel value such that

$$s_{ij}.w = s_{ji}.w = f\left(abs\left(n_i.p - n_j.p\right)\right), \tag{2}$$

where $s_{ij}.w$ is the weight of the synapse connecting neurons n_i and n_j, $f\left(\right)$ is a possibly nonlinear function and $n_i.p$ is neuron n_i's pixel value. The role of function $f\left(\right)$ is to create a strong connection between neurons with similar pixel values and a weak one between neurons with largely different pixel values. The function $f\left(\right)$ must be defined such that its value lies between 0 and 1. Examples are presented (section 2.2.5) with $f\left(\right)$ defined as

$$f\left(abs\left(n_i.p - n_j.p\right)\right) = \frac{MAX - abs\left(n_i.p - n_j.p\right)}{MAX}, \tag{3}$$

where MAX is the maximum possible pixel value.

2.2.4. *Iterative algorithm*

An iteration of the algorithm passes through the list of neurons and updates their state. The update at iteration k is computed as follows:

$$n_i.s\left(k\right) = \frac{n_i.s\left(k-1\right) + \sum_j n_j.s\left(k-1\right) \times s_{ij}.w}{NORM}, \tag{4}$$

where $n_i.s(k)$ is neuron n_i's state at iteration k. The constant $NORM$ is a factor, equals to 9 in our case, that normalizes the neurons state to 1. When this algorithm is run, neurons lying in the most highly textured regions (with high local gradients) of the image see their state variable drop very quickly. The neurons in homogeneous regions will maintain a state variable close to 1. After a given number of iterations, a thresholding is performed as follows:

$$n_i.s\left(k\right) \leftarrow \begin{cases} 0 & \text{if } n_i.s(k) \le THRESH \\ n_i.s\left(k\right) & \text{otherwise} \end{cases} \tag{5}$$

where $THRESH$ is the value of the threshold (in the experiments, a threshold value was empirically chosen to be around $\frac{1}{255}$). The thresholding is done to detect neurons lying in interesting contrasted regions of the image. The pixels associated with the thresholded neurons ($n_i.s(k) = 0$) are identified as being part of objects to be encoded into sounds in the subsequent steps of the algorithm.

2.2.5. *Object enhancement experiments*

Fig 3 illustrates enhancement on four images. Images have been first converted into grayscale levels. Then, gray levels are used as pixel values in equations 2 and 3 to compute the connecting weights. Then the algorithm has been iterated 7 times ($k = 7$).

The algorithm is able to enhance large objects (the car in images (c) and (d)) as well as small objects (the stake on the left of images (a) and (c)) and even the motorcyclist in images (a) and (b)). Furthermore, in more complex situations like the ones in images (e) and (g), the algorithm is able to enhance objects or groups of objects (the fish and the pills in image (f) or the group of person and the camera stand in image (h)). The enhanced images can then be used to generate sounds representing the visual scene for the substitution system.

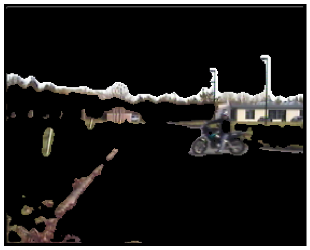
2.3. Sound generation

The sound generation has been developed based on human auditory features. Indeed, we are using Head Related Transfer Functions [23] (HRTF) to create a virtual acoustical space (VAS). Most stereophonic recordings are based only on differences in level between the two ears and the resulting sounds are therefore lateralized toward one ear or the other. By contrast, VAS stimuli incorporate the full complement of location cues and, in principle, replicate real free-field sounds.

The VAS technique involves two stages:

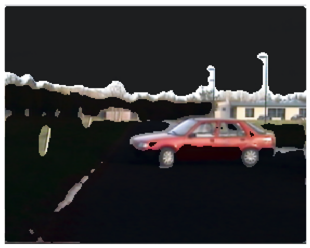
1. Recording the transfer function of the head: The ILDs (Interaural Level Difference), ITDs (Interaural Time Difference), and spectral cues make up what is known as the head-related transfer function (HRTF) which defines how the head and outer ears filter incoming sound. An accurate effect can be produced by measuring the precise HRTF of a patient. A foreign HRTF or an averaged HRTF taken over many listeners can still be used. Individual

(a) Reference image 1



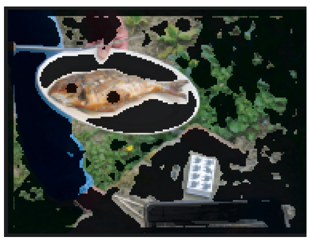
(b) Enhanced image 1

(c) Reference image 2



(d) Enhanced image 2

(e) Reference image 3



(f) Enhanced image 3

(g) Reference image 4

(h) Enhanced image 4

Fig. 3. Object enhancement: images on the left are the references and images on the right are the object enhanced images.

HRTFs are measured by placing miniature probe microphones into the subjects ears and recording the impulse responses to broad-band sounds presented from a range of directions in space.

2. Playing back the sounds through a VAS filter : the bank of HRTF impulse responses are now be converted into a filter bank. Any desired sound can be convolved with one of these filters and played to a listener over headphones. This creates the perception of an externalized sound source.

Using VAS, the sounds generated for a given object is convolved by one filter depending on its location in the image. As illustrated in figure 4, this allows the patient to locate the objects in front of him. It gives the impression that the object emits sound.

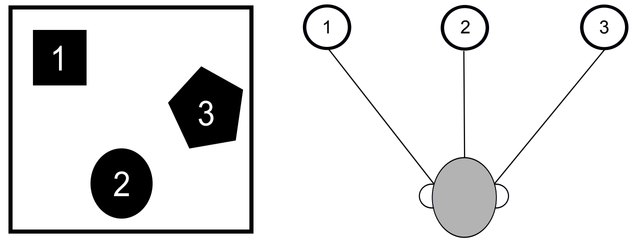


Fig. 4. Illustration of VAS principle: The rectangle on the left of the image represents limits of the visual field of the person. The first sound corresponding to the first object would appear to come from the top left corner of the visual field, whereas the second would appear to come from the bottom of the visual field and the third one would appear to come from the right of the visual field.

3. PERSPECTIVES AND DISCUSSION

3.1. Limits of the system

In this project, the main goal is to help blind people to perceive an overview of the visual scene in front of them so that they can walk around in a cluttered environment. As a first prototype, only the location of the objects is encoded into sounds. Future work involves the use of object recognition algorithms to increase the amount of information delivered to the patient.

3.2. Future issues

Another point which will add constraints is the real time factor. Indeed, for this system to be performant, the image processing and the sound generation will have to be extremely fast. At first, we will process image by image captured by the camera. Then we will accelerate to the maximum the image processing to be the closest of a real time system. The sound generation will not be a problem as it is already very fast.

The substitution system is intended to be worn by the patient, possibly on the head. Thus, the device will be subject to very sudden movements. The image processing algorithm will have to be impervious to this kind of external perturbation. This issue is not yet solved.

Finally, a question arises about the maximal amount of information from the visual scene that can be carried by the auditory pathway of the patient. As the system is developed, experiments will have to be done to determine, for instance, how many distinct objects can be easily located by the users of the system. Next steps of this project are first to generate simple sounds which will also be representative of the visual scene and pleasant to the ears. As this system will be tested by a lot of people, the use of a universal HRTF might be used as a solution to make the system more robust. Then we will think about the computing platform we will use (mobile device, FPGA...).

4. CONCLUSION

We have illustrated on real images how the object extraction algorithm is able to enhance small or large objects. We are using this approach to help in the extraction of important objects that are in the visual scene.

The strength of the proposed approach is the combination of an object-based image analysis with the sound generator so that mental visual representations can be carried via an auditory stimulation. Indeed, this system does not convert the entire image into sound but only parts corresponding to important features of the image. Furthermore, the sound generation is based on human auditory features like HRTFs. At first, this approach should help people with visual disability to move around in cluttered environments.

Acknowledgment

Thanks to FQRNT équipe and ACELP GEGI for funding, the ROBIN challenge [24] for providing images (a) and (c), Frédéric Mailhot for informal discussions on the neural enhancement algorithm and Stéphane Molotchnikoff for informal discussions on sensorial substitutions and Jan Schnupp for refeering us on HRTFs publications.

5. REFERENCES

[1] P Bach-Y-Rita, C C Collins, F A Saunders, B White, and L Scadden, "Vision substitution by tactile image projection," *Nature*, vol. 221, pp. 963–964, 1969.

[2] G Jansson, "Tactile guidance of movement," *International Journal of Neuroscience*, vol. 19, pp. 37–46, 1983.

[3] E Sampaio, S Maris, and P Bach-y Rita, "Brain plasticity: 'Visual' acuity of blind persons via the tongue," *Brain Research*, vol. 908, pp. 204–207, 2001.

[4] K A Kaczmarek and S J Haase, "Pattern identification and perceived stimulus quality as a function of stimulation current on a fingertip-scanned electrotactile display," *IEEE Transactions on Neural System Rehabilitation Engineering*, vol. 11, pp. 9–16, 2003.

[5] J C Bliss, M H Katcher, and C H Rogers, "Optical-to-tactile image conversion for the blind," *IEEE Transactions on Man - Machine Systems*, vol. 11, pp. 58–65, 1970.

[6] Takintope Akinbiyi, Carol E Reiley, Sunipa Saha, Darius Burschka, Christopher J Hasser, David D Yuh, and Allison M Okamura, "Dynamic augmented reality for sensory substitution in robot-assisted surgical systems.," *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, vol. 1, pp. 567–70, Jan. 2006.

[7] Lotfi B Merabet, Lorella Battelli, Souzana Obretenova, Sara Maguire, Peter Meijer, and Alvaro Pascual-Leone, "Functional recruitment of visual cortex for sound encoded object identification in the blind.," *Neuroreport*, vol. 20, no. 2, pp. 132–8, Jan. 2009.

[8] S. Hanneton, M. Auvray, and B. Durette, "The vibe: a versatile vision-to-audition sensory substitution device," *Applied Bionics and Biomechanics*, vol. 7, no. 4, pp. 269 –276, 2010.

[9] P.B.L. Meijer, "An experimental system for auditory image representations," *Biomedical Engineering, IEEE Transactions on*, vol. 39, no. 2, pp. 112 –121, 1992.

[10] C. Capelle, C. Trullemans, P. Arno, and C. Veraart, "A real-time experimental prototype for enhancement of vision rehabilitation using auditory substitution," *Biomedical Engineering, IEEE Transactions on*, vol. 45, no. 10, pp. 1279 –1293, 1998.

[11] J Cronly-Dillon, K Persaud, and R P Gregory, "The perception of visual images encoded in musical form: a study in cross-modality information transfer.," *Proceedings. Biological sciences / The Royal Society*, vol. 266, no. 1436, pp. 2427–33, Dec. 1999.

[12] J. Schnupp, I. Nelken, and A.J. King, *Auditory Neuroscience: Making Sense of Sound*, The MIT Press, 2011.

[13] I. Kokkinos and A. Yuille, "Hop: Hierarchical object parsing," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on DOI - 10.1109/CVPR.2009.5206639*, 2009, pp. 802–809.

[14] Michael I. Jordan (ed), *Learning in Graphical Models*, MIT Press, 1998.

[15] Maximilian Riesenhuber and Tomaso Poggio, "Hierarchical models of object recognition in cortex," *Nature neuroscience*, vol. 2, no. 11, pp. 1019–1025, 1999.

[16] Joel Pinto, Sivaram Garimella, Mathew Magimai-Doss, Hynek Hermansky, and Hervé Bourlard, "Analysis of MLP-Based Hierarchical Phoneme Posterior Probability Estimator," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 2, pp. 225–241, Feb. 2011.

[17] A. Leonardis S. Fidler, M. Boben, "Learning hierarchical compositional representations of object structure," in *Object Categorization: Computer and Human Vision Perspectives*, B. Schiele S. Dickinson, A. Leonardis and M. J. Tarr, Eds. Cambridge University Press, 2009.

[18] Gary Bradski and Adrian Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*, O'Reilly, 2008.

[19] P.M. Milner, "A model for visual shape recognition," *Psychological Review*, vol. 81, no. 6, pp. 521–535, Nov. 1974.

[20] Christoph Von Der Malsburg, "The correlation theory of brain function," *Models of Neural Networks II: Temporal Aspects of Coding and Information Processing in Biological Systems*, , no. July 1981, pp. 95–119, 1994.

[21] Louis-Charles Caron, Frèdèric Mailhot, and Jean Rouat, "FPGA implementation of a spiking neural network for pattern matching," in *IEEE Int. Symp. on Circuits and Systems (ISCAS)*, Rio de Janeiro, 15–18 May 2011, p. 1342.

[22] Vincent de Ladurantaye, Jean Lavoie, Maxime Parenteau, Jocelyn Bergeron, Huizhong Lu, Ramin Pichevar, and Jean Rouat, "A parallel supercomputer implementation of a biological inspired neural network and its use for pattern recognition," in *Proc. of High Perform. Comput. Symp. (HPCS)*, Montreal, 15–17 June 2011, Journal of Physics: Conference Series.

[23] V.R. Algazi, R.O. Duda, D.M. Thompson, and C. Avendano, "The cipic hrtf database," in *Applications of Signal Processing to Audio and Acoustics, 2001 IEEE Workshop on the DOI - 10.1109/ASPAA.2001.969552*, 2001, pp. 99–102.

[24] ROBIN Challenge, "Robin competition," http://robin.inrialpes.fr/.

ANNEXE B

Code algorithme de mise en valeur des objets

```
// Create the AVCapture Session
session = [[AVCaptureSession alloc] init];
```

Figure B.1 Création d'une AVCaptureSession.

```
// create a preview layer to show the output from the camera
AVCaptureVideoPreviewLayer *previewLayer = [AVCaptureVideoPreviewLayer layerWithSession:session];
previewLayer.frame = previewView.frame;
[previewView.layer addSublayer:previewLayer];
```

Figure B.2 Ajout de la vue de la caméra : la première ligne permet de créer un objet AVCaptureVideoPreviewLayer et de l'associer à la session créée précédemment. La deuxième ligne indique la trame que nous utilisons. La troisième ligne permet d'ajouter une couche à notre image de sortie.

```
// Get the default camera device
AVCaptureDevice* camera = [AVCaptureDevice defaultDeviceWithMediaType:AVMediaTypeVideo];

// Create a AVCaptureInput with the camera device
NSError *error=nil;
AVCaptureInput* cameraInput = [[AVCaptureDeviceInput alloc] initWithDevice:camera error:&error];
if (cameraInput == nil) {
    NSLog(@"Error to create camera capture:%@",error);
```

Figure B.3 Assignation de la caméra à utiliser : la première section de code permet de définir la caméra que l'on utilise (ici celle par défaut situé à l'arrière de l'appareil). La deuxième section de code assigne une entrée à la caméra sauf si une erreur se produit et dans ce cas-ci un message d'erreur est affiché.

```
// Set the output
AVCaptureVideoDataOutput* videoOutput = [[AVCaptureVideoDataOutput alloc] init];

// create a queue to run the capture on
dispatch_queue_t captureQueue=dispatch_queue_create("catpureQueue", NULL);

// setup our delegate
[videoOutput setSampleBufferDelegate:self queue:captureQueue];

// configure the pixel format
videoOutput.videoSettings = [NSDictionary dictionaryWithObjectsAndKeys:[NSNumber numberWithUnsignedInt:
kCVPixelFormatType_32BGRA], (id)kCVPixelBufferPixelFormatTypeKey,nil];

// and the size of the frames we want
[session setSessionPreset:AVCaptureSessionPresetLow];
```

Figure B.4 Configuration de la capture de la sortie.

```
// Add the input and output
[session addInput:cameraInput];
[session addOutput:videoOutput];

// Start the session
[session startRunning];
```

Figure B.5 Ajout de l'entrée et de la sortie à la session et départ de la simulation.

```
// only run if we're not already processing an image
if(imageToProcess==NULL) {
    // this is the image buffer
    CVImageBufferRef cvimgRef = CMSampleBufferGetImageBuffer(sampleBuffer);
    // Lock the image buffer
    CVPixelBufferLockBaseAddress(cvimgRef,0);
    // access the data
    int width=CVPixelBufferGetWidth(cvimgRef);
    int height=CVPixelBufferGetHeight(cvimgRef);
    // get the raw image bytes
    uint8_t *buf=(uint8_t *) CVPixelBufferGetBaseAddress(cvimgRef);
    size_t bprow=CVPixelBufferGetBytesPerRow(cvimgRef);
```

Figure B.6 Récupération des données de l'image à partir de la mémoire temporaire.

ANNEXE C
Optimisation de l'interface de l'application

Afin de permettre une meilleure utilisation et une meilleure compréhension de l'algorithme de mise en valeurs des objets, nous avons ajoutés les éléments suivant à l'interface de l'application :

- Un bouton pour modifier la valeur du seuil
- Un bouton pour activer ou désactiver le filtre médian
- Un bouton pour modifier le nombre d'itérations
- Un bouton pour modifier la résolution de l'image (haute ou basse résolution)
- Un label pour afficher le temps de calcul

C.1 Design de l'interface

Dans un premier temps, nos avons dû designer l'interface que nous souhaitions dans xcode. Pour cela nous avons ajouté différents boutons, du texte ainsi que des "label" afin d'afficher les valeurs de certaines variables (figure C.1).

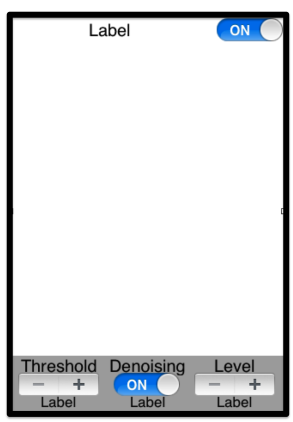


Figure C.1 Design de l'interface sous xcode : les boutons Threshold et Level servent à modifier la valeur du seuil et le nombre d'itération. Le bouton en haut permet de modifier la résolution de l'image et, celui en bas au milieu permet d'activer ou de désactiver le filtre médian. Le "label" en haut permet d'afficher la résolution sélectionnée, celui en bas à gauche la valeur du seuil, celui en bas au milieu le temps de calcul et enfin celui en bas à droite le nombre d'itérations.

Au niveau du graphisme, nous avons ajouté le logo Necotis comme logo pour l'application (figure C.2) et comme écran d'accueil (figure C.3).

Figure C.2 Page d'un Ipod Touch avec le logo de l'application.

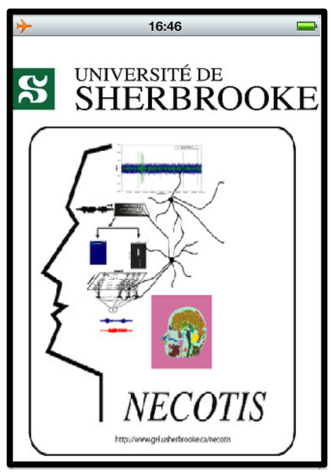


Figure C.3 Écran d'accueil de l'application : cet écran reste affiché pendant 2 secondes au lancement de l'application.

C.2 Modification de la valeur du seuil

Nous souhaitons ainsi rendre ajustable le seuil (THRESH) de l'équation (5) de l'annexe A. L'ajustement de ce seuil permet de contrôler les neurones considérés comme actif et ceux non-actifs. Pour cela, dans un premier temps, nous devons initialiser les valeurs maximales, les valeurs minimales et d'autres paramètres correspondants au bouton de modification du seuil (figure C.4).

```
self.stepper1Object.minimumValue = 0;
self.stepper1Object.maximumValue = 0.5;
self.stepper1Object.stepValue = 0.0001;
self.stepper1Object.wraps = YES;
self.stepper1Object.autorepeat = YES;
self.stepper1Object.continuous = YES;
```

Figure C.4 Initialisation des paramètres pour le bouton de modification du seuil : la valeur minimale du bouton est 0, sa valeur maximale est 0,5 et son pas d'avancement est de 0,0001. Les autres paramètres permettent de faire en sorte que la valeur s'incrémente si l'utilisateur reste appuyé sur le bouton.

Dans un second temps, nous devions implémenter la fonction qui est appelée à chaque fois que le bouton est pressé (figure C.5).

```
- (IBAction)stepper1:(id)sender
{
    tresh = stepper1Object.value;
    self.labelTresh.text = [NSString stringWithFormat:@"%.6f", stepper1Object.
        value];
}
```

Figure C.5 Implémentation de la fonction qui gère la modification du seuil : à chaque fois que le bouton est pressé, la variable du seuil *tresh* prend la valeur du bouton et sa valeur est affichée dans le "label" correspondant.

C.3 Activation ou désactivation du filtre médian

Pour ce bouton, nous stockons juste l'état du bouton dans la variable booléenne *filter* (figure C.6). Par la suite, la valeur de cette variable est testée et selon sa valeur le filtre est appliqué ou pas.

```
- (IBAction)filterValue:(id)sender {
    filter=filterButton.isOn;
}
```

Figure C.6 Implémentation de la fonction qui gère l'état du filtre : le filtre est initialisé comme actif et la variable *filter* est mise-à-jours à chaque action sur le bouton.

C.4 Modification du nombre d'itérations

L'implémentation de ce bouton a été effectuée comme celui du seuil, en deux étapes. Première étape, initialisation des paramètres du bouton (figure C.7).

```
self.stepper2Object.minimumValue = 0;
self.stepper2Object.maximumValue = 4;
self.stepper2Object.stepValue = 1;
self.stepper2Object.wraps = YES;
self.stepper2Object.autorepeat = YES;
self.stepper2Object.continuous = YES;
```

Figure C.7 Initialisation des paramètres pour le bouton de modification du nombre d'itérations : la valeur minimale du bouton est 0, sa valeur maximale est 4 et son pas d'avancement est de 1. Les autres paramètres permettent de faire en sorte que la valeur s'incrémente si l'utilisateur reste appuyé sur le bouton. Sachant que le compteur va de 0 à 4, ce qui fait en réalité de 1 à 5 itérations.

La deuxième étape consistait à implémenter la fonction qui était appelée à chaque fois que le bouton est pressé. Afin de permettre à l'utilisateur une navigation plus simple, nous avons pré-enregistré un seuil de référence pour chaque valeur possible du nombre d'itérations.

Ce pré-enregistrement permet à l'utilisateur de ne pas perdre de temps à chercher le seuil adéquat lorsque celui-ci change le nombre d'itérations (figure C.8).

```
- (IBAction)stepper2:(id)sender
{
    iteration = stepper2Object.value;
    self.labelFilter.text = [NSString stringWithFormat:@"%.f", stepper2Object.
        value+1];
    if (iteration==0) {
        stepper1Object.value=0.1120000;
        tresh = stepper1Object.value;
        self.labelTresh.text = [NSString stringWithFormat:@"%.6f", stepper1Object.
            value];
        self.stepper1Object.stepValue = 0.00001;
    }

    else if (iteration==1) {
        stepper1Object.value=0.0129015686;
        tresh = stepper1Object.value;
        self.labelTresh.text = [NSString stringWithFormat:@"%.6f", stepper1Object.
            value];
        self.stepper1Object.stepValue = 0.0001;
    }
    else if (iteration==2) {
        stepper1Object.value=0.010502;
        tresh = stepper1Object.value;
        self.labelTresh.text = [NSString stringWithFormat:@"%.6f", stepper1Object.
            value];
        self.stepper1Object.stepValue = 0.0001;
    }
    else if (iteration==3) {
        stepper1Object.value=0.002302;
        tresh = stepper1Object.value;
        self.labelTresh.text = [NSString stringWithFormat:@"%.6f", stepper1Object.
            value];
        self.stepper1Object.stepValue = 0.00001;
    }
    else if (iteration==4) {
        stepper1Object.value=0.000202;
        tresh = stepper1Object.value;
        self.labelTresh.text = [NSString stringWithFormat:@"%.6f", stepper1Object.
            value];
        self.stepper1Object.stepValue = 0.00001;
    }
}
```

Figure C.8 Implémentation de la fonction qui gère la modification du nombre d'itérations : à chaque fois que le bouton est pressé, la variable du nombre d'itérations *itération* prend la valeur du bouton et sa valeur est affichée dans le "label" correspondant. De plus, suivant le nombre d'itérations sélectionnée, la variable du seuil *tresh* prend la valeur pré-enregistrée pour ce nombre d'itérations et sa valeur est affichée dans le "label" correspondant. Suivant le nombre d'itérations sélectionné, le pas d'avancement de modification est modifié afin d'avoir plus ou moins de précision suivant la valeur du seuil.

C.5 Modification de la résolution de l'image

Pour ce bouton, nous stockons juste l'état du bouton dans la variable booléenne *quality*. Par la suite, la valeur de cette variable est testée et selon sa valeur la résolution de l'image est initialisée (figure C.9).

```
- (IBAction)actionQuality:(id)sender {
    quality=buttonQuality.isOn;
    if (quality==TRUE) {
        // and the size of the frames we want
        [session setSessionPreset:AVCaptureSessionPresetLow];
        self.labelQuality.text=[NSString stringWithFormat:@"Switch to High
            resolution"];
    }
    else {
        // and the size of the frames we want
        [session setSessionPreset:AVCaptureSessionPresetMedium];
        self.labelQuality.text=[NSString stringWithFormat:@"Switch to Low
            resolution"];
    }
}
```

Figure C.9 Implémentation de la fonction qui gère la résolution de l'image : la résolution est initialisée comme basse et la variable *quality* est mise-à-jour à chaque action sur le bouton.

La résolution basse est de 192x144 pixels et la résolution haute est de 480x360 pixels..

C.6 Affichage du temps de calcul

Le temps de calcul est mesuré à l'aide d'un chronomètre interne et sa valeur est stockée dans la variable *executionTime*. Par la suite sa valeur est affichée dans le "label" correspondant (figure C.10).

```
self.labelTime.text = [NSString stringWithFormat:@"%.f ms", executionTime*
    1000];
```

Figure C.10 Affichage du temps de calcul : la variable *executionTime* est multipliée par 1000 afin d'être affichée en milli-secondes.

ANNEXE D

Code de détection des zones actives

```
globalActivity++;
if (x>=0 && x<48 && y>=0 && y<64) {
    active1++;
}
else if (x>=48 && x<96 && y>=0 && y<64) {
    active2++;
}
else if (x>=96 && x<144 && y>=0 && y<64) {
    active3++;
}

else if (x>=0 && x<48 && y>=64 && y<128) {
    active4++;
}
else if (x>=48 && x<96 && y>=64 && y<128) {
    active5++;
}
else if (x>=96 && x<144 && y>=64 && y<128) {
    active6++;
}

else if (x>=0 && x<48 && y>=128 && y<192) {
    active7++;
}
else if (x>=48 && x<96 && y>=128 && y<192) {
    active8++;
}
else if (x>=96 && x<144 && y>=128 && y<192) {
    active9++;
}
```

Figure D.1 Calcul de l'activité globale et locale : la variable *globalActivity* correspond à l'activité globale et les variables *active*i correspondent à l'activité locale dans la zone i.

```
active1=active1/globalActivity;
active2=active2/globalActivity;
active3=active3/globalActivity;
active4=active4/globalActivity;
active5=active5/globalActivity;
active6=active6/globalActivity;
active7=active7/globalActivity;
active8=active8/globalActivity;
active9=active9/globalActivity;
```

Figure D.2 Calcul du taux d'activité par zone : les variables *active*i deviennent le taux d'activité dans la zone i.

```
//zone1
if (active1>activityThresh) {
    CGPathAddRect(pathRef, NULL,CGRectMake(0,0,48,64));
}
//zone2
if (active2>activityThresh) {
    CGPathAddRect(pathRef, NULL,CGRectMake(48,0,48,64));
}
//zone3
if (active3>activityThresh) {
    CGPathAddRect(pathRef, NULL,CGRectMake(96,0,48,64));
}
//zone4
if (active4>activityThresh) {
    CGPathAddRect(pathRef, NULL,CGRectMake(0,64,48,64));
}
//zone5
if (active5>activityThresh) {
    CGPathAddRect(pathRef, NULL,CGRectMake(48,64,48,64));
}
//zone6
if (active6>activityThresh) {
    CGPathAddRect(pathRef, NULL,CGRectMake(96,64,48,64));
}
//zone7
if (active7>activityThresh) {
    CGPathAddRect(pathRef, NULL,CGRectMake(0,128,48,64));
}
//zone8
if (active8>activityThresh) {
    CGPathAddRect(pathRef, NULL,CGRectMake(48,128,48,64));
}
//zone9
if (active9>activityThresh) {
    CGPathAddRect(pathRef, NULL,CGRectMake(96,128,48,64));
}
```

Figure D.3 Seuillage du taux d'activité et affichage des zones : la variable *activityThresh* est le seuil pour chaque zone (ici fixé à 25%) et la fonction *CG-PathAddRect* permet d'afficher un rectangle délimitant une zone.

ANNEXE E

Code génération de sons

```
y = pinknoise(100,5,44100);
```

Figure E.1 Fonction Matlab pinknoise qui permet de générer un bruit rose. Ici, le bruit rose généré a un gain de 100, dure 5 secondes et est à 44100 Hz.

```
%% config input and output
%Input = analoginput('nidaq','PXI2Slot3');
Input = analoginput('nidaq','Dev5');
addchannel(Input,0);
Input.Channel.InputRange = [-3.162278 3.162278];
Input.Channel.SensorRange = [-1 1];
Input.Channel.UnitsRange = [-1 1];
set(Input,'SampleRate',Fs)
set(Input,'SamplesPerTrigger',Fs*T)

Output = analogoutput('nidaq','Dev5');
addchannel(Output,0,{'PinkNoise'});
Output.Channel.OutputRange = [-1 1];
Output.Channel.UnitsRange = [-1 1];
set(Output,'SampleRate',Fs);
```

Figure E.2 Configuration des entrées et des sorties pour le système d'acquisition. Ici, une seule entrée est configurée. Pour ajouter la deuxième entrée, il suffit de remplacer la quatrième ligne par addchannel(Input,[0 1]).

```
%% Data acquisition
%load PXI data memory
putdata(Output,SOURCE)
%trigger input acquisition
start(Input);
%start playback
start(Output);
%timeout protection
wait(Output,T+2);
stop(Output);
%timout protection
wait(Input,T+2);
%fetch data
Input_data=getdata(Input,T*Fs);
```

Figure E.3 Lancement du système d'acquisition.

```
reference=load('C:\Documents and Settings\Administrateur\Bureau\damien\Resultats\2012_06_06\data_Matlab\reference
\20120606_reference_315_45.mat');

resultats=load('C:\Documents and Settings\Administrateur\Bureau\damien\Resultats\2012_06_06\data_Matlab\resultats
\20120606_resultats_315_45.mat');

resultatsG=resultats.Input_data(:,1);
resultatsD=resultats.Input_data(:,2);
[x,fs]=wavread('beep-1');
```

Figure E.4 Chargement des données Matlab et du son simple d'exemple.

```
% Now let's get the transfer function estimate, and calculate the
% filter coefficients using INVFREQZ.
%left ear
[tgxy,wg] = tfestimate(reference.Input_data,resultatsG);
[bg,ag] = invfreqz(tgxy,wg,100,3);
sortieG=filter(bg,ag,x);
%right ear
[trxy,wr] = tfestimate(reference.Input_data,resultatsD);
[br,ar] = invfreqz(trxy,wr,100,3);
sortieR=filter(br,ar,x);
```

Figure E.5 Estimation de la fonction de transfert, calcul de la réponse impulsionnelle et filtrage du son pour chaque oreille.

```
%sound creation
sound=[sortieG,sortieR];
wavwrite(sound,fs,'sortie');
```

Figure E.6 Filtrage du son et enregistrement du son filtré.

```
NSURL *url1 = [NSURL fileURLWithPath:[NSString stringWithFormat:@"%@/sortie1.
    wav", [[NSBundle mainBundle] resourcePath]]];
audioPlayer1 = [[AVAudioPlayer alloc] initWithContentsOfURL:url1 error:&error1
    ];
audioPlayer1.numberOfLoops = -1;
audioPlayer1.volume = 0.5;
```

Figure E.7 Initialisation des sons : chaque son est chargé avec son type de fichier.

```
[audioPlayer1 play];
```

Figure E.8 Activation des sons : si la zone correspondant au son est active alors le son est lancé.

```
[audioPlayer1 stop];
```

Figure E.9 Arrêt des sons : tous les sons en cours sont arrêtés avant de relancer ceux pour lesquels leur zone est active.

```
audioPlayer1.currentTime=0.2;
```

Figure E.10 Remise à zéro des sons : tous les sons remis à zéro avant d'être relancés.

LISTE DES RÉFÉRENCES

Akinbiyi, T., Reiley, C. E., Saha, S., Burschka, D., Hasser, C. J., Yuh, D. D. et Okamura, A. M. (2006). Dynamic augmented reality for sensory substitution in robot-assisted surgical systems. *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society.*, volume 1, p. 567–70.

Algazi, V., Duda, R., Thompson, D. et Avendano, C. (2001). The CIPIC HRTF database. Dans *IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics*. p. 99 –102.

Amedi, A., Stern, W. M., Camprodon, J. A., Bermpohl, F., Merabet, L., Rotman, S., Hemond, C., Meijer, P. et Pascual-Leone, A. (2007). Shape conveyed by visual-to-auditory sensory substitution activates the lateral occipital complex. *Nature neuroscience*, volume 10, numéro 6, p. 687–9.

Apple (2012). Siège social : Infinite loop, cupertino, californie (États-Unis). `https://developer.apple.com/devcenter/ios/index.action`.

Auvray, M., Hanneton, S. et O'Regan, J. K. (2007). Learning to perceive with a visuo - auditory substitution system : Localisation and object recognition with "The vOICe". *Perception*, volume 36, numéro 3, p. 416–430.

Bach-Y-Rita, P. (2003). Sensory substitution and the human-machine interface. *Trends in Cognitive Sciences*, volume 7, numéro 12, p. 541–546.

Bach-Y-Rita, P., Collins, C. C., Saunders, F. A., White, B. et Scadden, L. (1969). Vision substitution by tactile image projection. *Nature*, volume 221, p. 963–964.

BlackBerry (2012). Siège social : Waterloo (ontario) (canada). `http://www.rim.com`.

Blauert, J. (1983). Spatial hearing : the psychophysics of human sound localization.

Bliss, J. C., Katcher, M. H. et Rogers, C. H. (1970). Optical-to-tactile image conversion for the blind. *IEEE Transactions on Man - Machine Systems*, volume 11, p. 58–65.

Brungart, D. S., Durlach, N. I. et Rabinowitz, W. M. (1999). Auditory localization of nearby sources. ii. localization of a broadband source. *J. Acoust. Soc. Am.*, volume 106, numéro 4, p. 1956–1968.

Canny, J. F. (1983). Finding edges and lines in images. *M.S. Thesis Inst. of Tech., Cambridge. Artificial Intelligence Lab.*

Capelle, C., Trullemans, C., Arno, P. et Veraart, C. (1998). A real-time experimental prototype for enhancement of vision rehabilitation using auditory substitution. *Biomedical Engineering, IEEE Transactions on*, volume 45, numéro 10, p. 1279 –1293.

Caron, L.-c., Mailhot, F. et Rouat, J. (2011). FPGA implementation of a spiking neural network for pattern matching. Dans *IEEE International Symposium on Circuits and Systems (ISCAS)*. p. 649–652.

Chun, M. M. et Wolfe, J. M. (2001). *Blackwell Handbook of Sensation and Perception*. Blackwell Publishers, 272–310 p.

Critias (2012). `http://critias.etsmtl.ca/arp`.

Cronly-Dillon, J., Persaud, K. et Gregory, R. P. (1999). The perception of visual images encoded in musical form : a study in cross-modality information transfer. *Proceedings. Biological sciences / The Royal Society*, volume 266, numéro 1436, p. 2427–33.

Cronly-Dillon, J., Persaud, K. C. et Blore, R. (2000). Blind subjects construct conscious mental images of visual scenes encoded in musical form. *Proceedings. Biological sciences / The Royal Society*, volume 267, numéro 1458, p. 2231–8.

Daniel-Audio-Labs (2012). `http://www.danielaudiolabs.com`.

de Ladurantaye, V., Lavoie, J., Bergeron, J., Parenteau, M., Lu, H., Pichevar, R. et Rouat, J. (2012). A parallel supercomputer implementation of a biological inspired neural network and its use for pattern recognition. *Journal of Physics : Conference Series*, volume 341, numéro 1, p. 012024.

Deatherage, B. H. et Hirsh, I. J. (1959). Auditory localization of clicks. *J. Acoust. Soc. Am.*, volume 31, numéro 4, p. 486–492.

Feddersen, W. E., Sandel, T. T. et Teas, D. C. (1957). Localization of high-frequency tones. *Journal of The Acoustical Society of America*, volume 29.

Feder, T. (2012). Shhhh. listen to the data. *PhysicsToday*, p. 20.

Frintrop, S., Klodt, M. et Rome, E. (2007). A real-time visual attention system using integral images. *Science*.

Gardner, M. B. (1968). Proximity image effect in sound localization. *The Journal of the Acoustical Society of America*, volume 43, numéro 1, p. 163–163.

Gartner (2012). `http://www.gartner.com/it/page.jsp?id=1466313`.

Google (2012). Siège social : Googleplex, mountain view (États-Unis). `http://www.android.com`.

G.R.A.S (2012). `http://ansihead.com/`.

Hanneton, S., Auvray, M. et Durette, B. (2010). The Vibe : a versatile vision-to-audition sensory substitution device. *Applied Bionics and Biomechanics*, volume 7, numéro 4, p. 269 –276.

Hartmann, W. et Wittenberg, A. (1996). On the externalization of sound images. *J. Acoust. Soc. Am.*, volume 99, numéro 6, p. 3678–3688.

Hughes, B. (2001). Active artificial echolocation and the nonvisual perception of aperture passability. *Human Movement Science*, volume 20, numéro 4-5, p. 371 – 400.

IMS-Research (2012). `http://imsresearch.com`.

Itti, L. (2000). *Models of Bottom-Up and Top-Down Visual Attention.* Thèse de doctorat, California Institute of Technology, 216 p.

Jansson, G. (1983). Tactile guidance of movement. *International Journal of Neuroscience*, volume 19, p. 37–46.

Junius, D., Riedel, H. et Kollmeier, B. (2007). The influence of externalization and spatial cues on the generation of auditory brainstem responses and middle latency responses. *Hearing Research*, volume 225, numéro 1-2, p. 91 – 104.

Kaczmarek, K. A. et Haase, S. J. (2003). Pattern identification and perceived stimulus quality as a function of stimulation current on a fingertip-scanned electrotactile display. *IEEE Transactions on Neural System Rehabilitation Engineering*, volume 11, p. 9–16.

Kay, L. (1964). An ultrasonic sensing probe as a mobility aid for the blind. *Ultrasonics*, volume 2, numéro 2, p. 53 – 59.

Kokkinos, I. et Yuille, A. (2009). Hop : Hierarchical object parsing. *IEEE Conference on Computer Vision and Pattern Recognition (2009)*, volume 1, p. 802–809.

Kokovay, E., Shen, Q. et Temple, S. (2008). Perspective The Incredible Elastic Brain : How Neural Stem Cells Expand Our Minds. *Neuron*, volume 60, numéro 3, p. 420–429.

Mathworks (2012). Siège social : Natick, massachusetts (États-Unis). `http://www.mathworks.com/products/matlab/?s_cid=wiki_matlab_2`.

Meijer, P. (1992). An experimental system for auditory image representations. *Biomedical Engineering, IEEE Transactions on*, volume 39, numéro 2, p. 112 –121.

Merabet, L. B., Battelli, L., Obretenova, S., Maguire, S., Meijer, P. et Pascual-Leone, A. (2009). Functional recruitment of visual cortex for sound encoded object identification in the blind. *Neuroreport*, volume 20, numéro 2, p. 132–8.

Microsoft (2012). Siège social : Redmond, washington (États-Unis). `http://www.microsoft.com/windowsphone`.

Milner, P. M. (1974). A model for visual shape recognition. *Psychological Review*, volume 81, numéro 6, p. 521–535.

Montandon, A. (2004). Colourblind eyeborg colours to sound. `http://www.adamç.com/neil-harbisson-the-cyborg/`.

National-Instruments (2012). Siège social : Austin (États-Unis). `http://www.ni.com/labview/f/`.

Nokia (2012). Siège social : Keilaniemi, espoo (finlande). `http://www.developer.nokia.com/Devices/Symbian/`.

Pantofaru, C. (2005). A comparison of image segmentation algorithms. *Robotics*, volume 18, p. 673 ?689.

Pascual-Leone, A., Amedi, A., Fregni, F. et Merabet, L. B. (2005). The plastic human brain cortex. *Annual review of neuroscience*, volume 28, p. 377–401.

Peters, C. et O'Sullivan, C. (2003). Bottom-up visual attention for virtual human animation. Dans *In Computer Animation for Social Agents*. p. 111–117.

Pinto, J., Garimella, S., Magimai-Doss, M., Hermansky, H. et Bourlard, H. (2011). Analysis of mlp-based hierarchical phoneme posterior probability estimator. *Ieee Transactions On Audio Speech And Language Processing*, volume 19, numéro 2, p. 225–241.

Pollok, B., Schnitzler, I., Stoerig, P., Mierdorf, T. et Schnitzler, A. (2005). Image-to-sound conversion : experience-induced plasticity in auditory cortex of blindfolded adults. *Experimental brain research. Experimentelle Hirnforschung. Expérimentation cérébrale*, volume 167, numéro 2, p. 287–91.

Renier, L., Collignon, O., Poirier, C., Tranduy, D., Vanlierde, A., Bol, A., Veraart, C. et Volder, A. G. D. (2005a). Cross-modal activation of visual cortex during depth perception using auditory substitution of vision. *NeuroImage*, volume 26, numéro 2, p. 573 – 580.

Renier, L., Laloyaux, C., Collignon, O., Tranduy, D., Vanlierde, a., Bruyer, R. et Volder, a. G. D. (2005b). The Ponzo illusion with auditory substitution of vision in sighted and early-blind subjects. *Perception*, volume 34, numéro 7, p. 857–867.

Riesenhuber, M. et Poggio, T. (1999). Hierarchical models of object recognition in cortex. *Nature Neuroscience*, volume 2, numéro 11, p. 1019–1025.

Rouat, J., Bergeron, J., Caron, L.-C., de Ladurantaye, V. et Mailhot, F. (June 2011). Method and device for providing structural representations of physical entities. US provisional patent application 61/493,672.

Ryan, K., Amer, A. et Gagnon, L. (2009). Spatiotemporal region enhancement and merging for unsupervized object segmentation. *EURASIP Journal on Image and Video Processing*, p. 1–13.

Sampaio, E., Maris, S. et Bach-y Rita, P. (2001). Brain plasticity : 'Visual' acuity of blind persons via the tongue. *Brain Research*, volume 908, p. 204–207.

Schnupp, J. (2011). `http://mustelid.physiol.ox.ac.uk/drupal/?q=topics/VAS`.

Schnupp, J., Nelken, I. et King, A. (2011). *Auditory Neuroscience : Making Sense of Sound*. The MIT Press.

Sculptedeers (2012). `http://sculptedeers.com`.

SONOMAX (2012). Siège social : Montreal (quebec) (canada). `http://sonomax.com/en/about-us.html`.

Strutt, J. (1907). On our perception of sound direction. *Philosophical Magazine*.

Summerhill, R. (2012). `http://aptogo.co.uk/2011/09/opencv-framework-for-ios/`.

Tianshu, Q., Zheng, X., Mei, G., Ying, H., Xiaodong, L. et Xihong, W. (2009). Distance-dependent head-related transfer functions measured with high spatial resolution using a spark gap. *IEEE Transactions on Audio, Speech and Language Processing*, p. 1124–1132.

Tukey, J. W. (1971). Exploratory data analysis.

Von Der Malsburg, C. (1981). The correlation theory of brain function. Springer-Verlag, p. 1–26.

Wenzel, E., Arruda, M., Kistler, D. et Foster, S. (1993). Localization using non-individualized head-related transfer funtions. *J. Acoust. Soc. Am.*, volume 94, p. 111–123.

Wolfe, J. M. (2000). Visual attention. *Frontiers in Bioscience*, volume 5, numéro 1, p. 335–386.