MODELS AND ALGORITHMS FOR INTERACTIVE AUDIO RENDERING

Nicolas Tsingos

UNIVERSITÉ DE NICE - SOPHIA ANTIPOLIS
ECOLE DOCTORALE STIC

MÉMOIRE

pour obtenir le grade

HABILITATION A DIRIGER DES RECHERCHES

Spécialité : Sciences pour l'ingénieur

Présentée et soutenue publiquement
par

Nicolas TSINGOS

Le 14 Avril 2008

MODELS AND ALGORITHMS FOR INTERACTIVE AUDIO RENDERING

COMPOSITION DU JURY :

Pierre	COMMON	Président
Durand	BEGAULT	Rapporteur
Davide	ROCCHESSO	Rapporteur
Karlheinz	BRANDENBURG	Rapporteur
George	DRETTAKIS	Examinateur
Olivier	WARUSFEL	Examinateur

Remerciements

Contents

Chapter 1

Introduction

Les systèmes de réalité virtuelle interactifs combinent des représentations visuelle, sonore et haptique, afin de simuler de manière immersive l'exploration d'un monde tridimensionnel représenté depuis le point de vue d'un observateur contrôlé en temps réel par l'utilisateur. La plupart des travaux effectués dans ce domaine ont historiquement porté sur les aspects visuels (par exemple des méthodes d'affichage interactif de modèles 3D complexes ou de simulation réaliste et efficace de l'éclairage) et relativement peu de travaux ont été consacrés à la simulation de sources sonores virtuelles également dénomée auralisation. Il est pourtant certain que la simulation sonore est un facteur clé dans la production d'environnements de synthèse, la perception sonore s'ajoutant à la perception visuelle pour produire une interaction plus naturelle. En particulier, les effets sonores spatialisés [Beg94], dont la direction de provenance est fidèlement reproduite aux oreilles de l'auditeur, sont particulièrement importants pour localiser les objets, séparer de multiples signaux sonores simultanés et donner des indices sur les caractéristiques spatiales de l'environnement (taille, matériaux, etc.). La plupart des systèmes de réalité virtuelle immersifs, des simulateurs les plus complexes aux jeux vidéo destinés au grand public mettent aujourd'hui en uvre des algorithmes de synthèse et spatialisation des sons qui permettent d'améliorer la navigation et d'accroître le réalisme et la sensation de présence de l'utilisateur dans l'environnement de synthèse. Comme la synthèse d'image dont elle est l'équivalent auditif, l'auralisation, appelée aussi rendu sonore, est un vaste sujet à la croisée de multiples disciplines : informatique, acoustique et électro-acoustique, traitement du signal, musique, calcul géométrique mais également psycho-acoustique et perception audio-visuelle. Elle regroupe trois problématiques principales: synthèse et contrôle interactif de sons, simulation des effets de propagation du son dans l'environnement et enfin, perception et restitution spatiale aux oreilles de l'auditeur. Historiquement, ces trois problématiques émergent de travaux en acoustique architecturale, acoustique musicale et psycho-acoustique. Toutefois une différence fondamentale entre rendu sonore pour la réalité virtuelle et acoustique réside dans l'interaction multimodale et dans l'efficacité des algorithmes devant être mis en œuvre pour des applications interactives. Ces aspects importants contribuent à en faire un domaine à part qui prend une importance croissante, tant dans le milieu de l'acoustique que dans celui de la synthèse d'image/réalité virtuelle.

1.1 Problématiques du rendu sonore

La première étape du pipeline de rendu sonore est la synthèse des sons qui doivent être émis par les sources virtuelles. Nos travaux n'apportent pas de contribution directe à ces aspects. Néanmoins, nous en proposons un rapide état de l'art dans le Chapitre 2.

Une fois synthétisés, les signaux sonores émis par les sources virtuelles sont ensuite traités de

manière à reproduire les effets de propagation du son dans l'environnement de synthèse (occultation par les obstacles, réflexions sur les parois et réverbération, effet Doppler pour les sources en mouvement). Ces effets dépendent fortement de la géométrie de l'environnement, de la position des sources et de l'auditeur. Les traitements effectués sont donc contrôlés par des calculs géométriques très proches de ceux effectués pour la synthèse d'image (du lancer de rayon par exemple) mais ont également leurs particularités comme le traitement de la diffraction du son par les obstacles qui est généralement négligeable pour la lumière. Pour les applications où le réalisme plus que la précision physique de la simulation est suffisant, des modèles perceptifs peuvent être également utilisés pour décrire les traitements à appliquer aux sons sans recourir à des simulations géométriques trop coûteuses en temps de calcul. Nous proposons un tour d'horizon de ces différentes approches dans le Chapitre 2 de ce document.

Enfin, la dernière etape du processus est la restitution du son aux oreilles de l'auditeur de manière la plus fidèle possible. Au delà des notions de fidélité généralement admises en audio (précision des calculs et de la représentation numérique des signaux, qualité de l'électronique et des hauts-parleurs), l'acoustique virtuelle s'intéresse tout particulièrement à la fidélité de la restitution spatiale du son, la spatialisation. La spatialisation du son est un traitement spécifique qui vise à reproduire aux oreilles de l'auditeur la sensation correcte de la direction de provenance des sons virtuels. C'est l'un des composants les plus importants dans la production de son pour la réalité virtuelle. On peut le comparer à la production d'images en relief pour le canal visuel. Depuis la mise au point des premiers systèmes de restitution stéréophonique et multi-canal dans les années 1930-1940, la reproduction spatiale du son a été un enjeu majeur de la recherche en acoustique et électro-acoustique. Longtemps cantonné aux laboratoires de recherche ou aux amateurs de musique expérimentale, le son spatial à récemment effectué une percée fulgurante chez le grand public au travers du son 3D ou surround (limité dans le plan horizontal) pour les jeux vidéo et le "home-cinéma". De nombreux systèmes de restitution sonore spatiale ont été développés et étudiés durant les 30 dernières années. La référence reste la stéréo binaurale qui permet de reproduire à l'aide d'un simple casque stéréo et d'un filtrage spécifique des signaux, la direction de provenance d'un son synthétisé ou enregistré. Ce système permet de reproduire aussi bien des sons provenant de l'avant que de l'arrière, du dessus ou du dessous de l'auditeur. Néanmoins, la qualité du résultat est très dépendante de l'auditeur puisque les filtres à appliquer sont fortement dépendants de la morphologie de la personne. D'autres systèmes comme l'holophonie, équivalent acoustique de l'holographie, multiplient le nombre d'enceintes utilisées pour reproduire le plus fidèlement possible les ondes sonores physiquement correctes dans le local d'écoute. La encore, bien que nos travaux ne portent pas directement sur le développement de nouvelles techniques de restitution, nous donnerons un aperçu de ces dispositifs dans le Chapitre 3 de ce document car ils jouent un rôle essentiel sur l'ensemble du pipeline de rendu sonore.

1.2 Contributions et structure du mémoire

Au fil des deux premiers chapitres de ce document, on notera souvent une dualité entre techniques basées sur la physique et qui se rapprochent de la simulation numérique et d'autres se contentant de reproduire l'effet perceptif souhaité de manière réaliste. Nos travaux s'articulent également autour de ces deux aspects.

Notre première contribution est dédiée aux approches de propagation du son par modèles physiques dans le but de les rendre plus interactives sans sacrifier à la qualité du résultat. Nous proposons en particulier une extension aux algorithmes de lancer de faisceaux afin d'inclure les phénomènes de diffraction qui peuvent jouer un rôle perceptif majeur. Nous proposons une evaluation de l'approximation de Kirchhoff pour aller au delà des techniques géométriques à base de propagation de rayons sonores et se rapprocher de simulations de type éléments finis, gérant les phénomènes ondulatoires. Nous montrerons

que ces approches permettent d'obtenir une modélisation fine de la réflexion du son pour des surfaces à la géometrie très complexe comprenant des millions de polygones. Enfin, nous montrerons que ces techniques de simulation peuvent tirer partie des ressources de calcul massives offertes par les cartes graphiques de nouvelle génération, tant pour les calculs de réflexion du son, que pour le traitement du signal requis pour pouvoir en apprécier auditivement le résultat.

Des optimisations alternatives, s'appuyant sur des aspects perceptifs et permettant de traiter des scènes très complexes, comprenant des centaines voire milliers de sources sonores, sont également au coeur de nos travaux. En particulier, nous proposons des techniques de spatialisation sonore progressives. Ces techniques, au croisement de la psycho-acoustique, de la compression audio perceptive (proches du standard *mp3* par des aspects d'évaluation de masquage auditif) et de l'analyse de scène sonore, introduisent des concepts de rendu à différents niveaux de détail, similaires à ceux développés pour le rendu graphique. Ces travaux feront l'objet du Chapitre 5.

Enfin, une dernière famille d'approches, alternative au rendu de sources sonores individuelles, peut être mise en parallèle avec le rendu à base d'image en graphique. Ces approches permettent une synthèse sonore spatiale directement à partir d'enregistrements ou de représentations plus structurées des sources sonores que l'on peut en extraire. Par exemple, elle permettent, grâce à des dispositifs de prises de son spécifiques, de générer des déplacements virtuels en travaillant directement sur un ensemble d'enregistre–ments simultanés d'un environnement réel réalisés en des positions différentes. Nos travaux dans ce domaine, feront l'objet du dernier chapitre de ce mémoire.

Chapter 2

Models for audio rendering

Ce chapitre propose un tour d'horizon des différentes modèles pouvant être mis en uvre pour apporter une dimension sonore immersive à un environnement virtuel interactif. Les sections 2.2.2, 2.3 et 2.4 couvrent également nos travaux propres, en les présentant brièvement dans le contexte plus général de l'état de l'art du domaine. Une partie en sera présentée plus en détails dans les chapitres suivants.

Ce chapitre est basé sur les articles:

Nicolas Tsingos et Olivier Warusfel.
Traité de la Réalité Virtuelle.
Tome II. Chapitre 15 - Dispositifs et interfaces de restitution sonore spatiale.
Tome III. Chapitre 4 - Modèles pour le rendu sonore.
Presses de l'Ecole des Mines de Paris, 2006.

En complément on pourra se référer aux ouvrages et articles suivants [Beg94, KDS93, SHLV99].

2.1 Synthesizing virtual sound sources

La première étape du pipeline de rendu sonore est la synthèse des sons qui doivent être émis par les sources virtuelles. La synthèse sonore est un vaste domaine qui a fait l'objet d'intenses recherches en particulier pour des applications musicales [Roa96, Coo02, PB04a]. Néanmoins, une particularité des environnements virtuels est que les sources sonores devant être synthétisées sont très rarement musicales. De nombreuses approches originales ont donc été proposées dans ce domaine afin de synthétiser de manière réaliste l'immense variété de sources sonores "naturelles" pouvant être rencontrées. De plus, la synthèse doit pouvoir être effectuée de manière à autoriser une interaction avec les actions de l'utilisateur ou à réagir automatiquement en fonction des interactions entre objets (les collisions par exemple). Dans ce contexte, on peut distinguer deux grandes familles d'approche: celles utilisant directement des enregistrements et celles synthétisant entièrement le signal sonore à partir de modèles mathématiques et physiques. Comme nous le verrons, des approches hybrides, combinant analyse d'enregistrements et re-synthèse permettent également de combiner au mieux réalisme et possibilité d'interaction avec le modèle.

2.1.1 Sample-based synthesis and sound textures

Une première approche pour la synthèse des signaux émis par les sources est l'utilisation directe d'enregi–strements correspondants (en anglais, *sampling*). Un ou plusieurs enregistrements, en général monophoniques, peuvent être combinés pour recréer le son voulu en fonction de différents paramètres de l'application. Par exemple, dans un simulateur ou jeu vidéo récent de conduite automobile, le son de chaque véhicule résulte de la combinaison de plusieurs dizaines d'enregistrements (bruit du moteur à différent régimes, bruit des pneumatiques, bruit aérodynamique, etc.). La combinaison de ces différents enregistrements est alors liée à des paramètres de plus haut niveau, par exemple des données physiques de la simulation (la vitesse, le régime moteur, etc.). Différents effets (modification de la hauteur, modification du spectre des sons) peuvent être également appliqués à ce stade en fonction de l'effet souhaité. L'approche à base d'enregistrements réels donne en général des résultats très réalistes mais au prix d'une prise de son qui peut s'avérer très complexe et d'un travail d'analyse (éventuellement manuel) important pour relier les combinaisons de sons aux différents paramètres de synthèse. De plus, elle est également assez lourde en ressources puisque les enregistrements peuvent occuper une place mémoire importante. Néanmoins, ce problème tend à disparaitre avec l'apparition de solutions matérielles permettant de décompresser à la volée les signaux audio au moment de leur utilisation.

Malgré cela, il est impossible d'enregistrer des signaux audio temporellement infinis. Les signaux continus (les bruits de moteur par exemple) doivent en pratique être re-bouclés sur eux-mêmes, ce qui peut conduire à une inévitable sensation de répétition qu'il est très difficile d'éliminer. Cette problématique a donné naissance à diverses techniques de synthèse de textures sonores, de manière similaire au graphique 3D [LWZ04, PC03, PB04b, SAP95, AE03, DS03]. A partir d'un son cible, on s'intéresse ici à synthétiser un signal similaire, non-répétitif et de longueur quelconque. Plusieurs approches peuvent être utilisées dans ce but. Les approches concaténatives segmentent le signal cible en plusieurs sous-parties en utilisant des métriques de similarité (Figure 2.1). Les possibilités de transition entre les différentes sous-parties sont alors évaluées pour construire un graphe de transition [LWZ04, Jeh05]. A partir de ce graphe, un signal infini peut être resynthétisé simplement en concaténant une partie à l'autre en fonction des probabilités de transition. D'autres techniques proposent une analyse statistique multi-échelle du signal de départ (par exemple en le décomposant en ondelettes) [DBJEY$^+$02]. D'autres utilisent des modèles paramétriques adaptés aux paramètres statistiques du signal cible [DCH00, BJLW$^+$99].

Enfin, une dernière difficulté liée à la synthèse par enregistrement est que ceux-ci doivent être les plus "bruts" possible, c'est à dire avec le minimum de coloration ou effets ds à l'environnement si on veut ensuite pouvoir leur appliquer des effets de propagation. Pour cela, il est nécessaire d'enregistrer avec des microphones très directifs et en champ proche de manière à obtenir un rapport signal-sur-interférence (i.e., par rapport au bruit de fond ou à la réverbération) satisfaisant. Une autre solution passe par des enregistrements dans une chambre anéchoque ou quasi-anéchoique, dans laquelle les parois sont traitées pour éliminer ou limiter les réflexions parasites.

2.1.2 Physically-based synthesis

La plupart des approches pour la synthèse physique de sons dans les environnements virtuels se concentre sur les bruits liés aux interactions entre objets (choc, roulement, frottement), qui à eux seuls constituent une vaste catégorie d'évènements sonores [MAB$^+$03]. Qui plus est, cette catégorie est fondamentale pour les environnements virtuels puisqu'elle permet de rendre audibles les interactions de l'utilisateur avec l'environnement. Ces approches se basent généralement sur une estimation des modes de vibration des objets puis par une étape de synthèse modale [DP98, vdDKP01, vdDPA$^+$02, vdDKP04, OSG02], le son étant représenté comme une somme de sinusoides amorties dans le temps. Les fréquences, ampli-

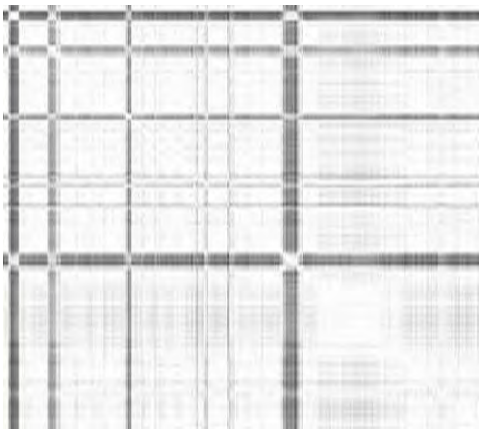


Figure 2.1: Visualisation de la similarité entre différentes trames successives d'un signal musical. L'intensité croit avec la similarité. De telles matrices de similarité intra-signal sont utilisées pour resynthétiser des textures sonores infinies, non répétitives.

tudes et décroissances des différents modes constituent les paramètres de la réponse impulsionnelle de l'objet. Cette réponse varie en fonction de la géométrie de l'objet, du matériau mais également du point d'impact et de la force de contact. Le son émis par l'objet dépend également au final de l'excitation. Dans le cas d'un choc, la réponse impulsionnelle peut être directement utilisée. Pour un frottement, il faut convoluer cette réponse par une représentation de l'excitation [vdDKP01]. Dans le cadre d'objets virtuels rigides, il est possible de pré-calculer la matrice des modes de vibrations en utilisant un maillage 3D de l'objet [OSG02]. Pour des objets déformables, la synthèse nécessite des calculs plus complexes à base d'éléments finis qui ne permettent pas encore des applications temps-réel [OCE01]. Enfin, une synthèse correcte nécessiterait également de simuler un couplage mécanique/acoustique, les vibrations de la surface de l'objet se propageant dans l'air et intéragissant avec sa propre géométrie (par exemple, pour prendre en compte la diffraction due à l'objet lui-même). Une alternative aux techniques de synthèse pure consiste à combiner analyse d'enregistrements et resynthèse. Par exemple, une approche permettant de mesurer la réponse acoustique d'objets réels a été présentée dans [vdDKP01]. Un bras robotique muni d'une pointe rigide est utilisé pour exciter la surface d'un objet dont la réponse acoustique est enregistrée par un microphone. Par échantillonnage de la surface de l'objet, on peut alors construire une texture 2D représentant la réponse impulsionnelle de l'objet en différents points de sa surface. L'analyse des réponses enregistrées permet une extraction des paramètres des principaux modes de vibration qui permettent alors une resynthèse des bruits de contact dans le cadre d'une interaction temps-réel avec une maquette virtuelle de l'objet. En particulier, ces approches se prêtent bien à une intégration avec une restitution haptique des contacts. D'autres types de synthèse ont également été proposés pour des phénomènes naturels comme les bruits aerodynamiques [DYN03] (vent, siffiement d'un coup d'épée) ou les bruits de combustion et explosion [DYN04]. Dans ce cas, une simulation de dynamique des fiuides par éléments finis est utilisée pour générer les paramètres de synthèse (vitesse du fiuide, etc.). Le son correspondant est alors synthétisé en sommant des textures sonores (en général du bruit blanc), modulées par les paramètres appropriés, pour chaque cellule de l'espace utilisée pour la simulation. On peut donc considérer cette approche comme une approche hybride entre synthèse purement physique et synthèse par enregistrements. Enfin, on peut également mentionner ici les travaux de Van Den Doel [Doe04] qui utilisent un modèle simplifié pour synthétiser le bruit créé par la formation d'une bulle d'air dans un liquide et le combine avec des modèles statistiques pour synthétiser des combinaisons plus complexes,

évoquant des bruits de pluie ou de torrent. Pour plus d'information concernant des travaux récents en synthèse sonore, nous renvoyons également le lecteur aux travaux effectués dans le cadre du projet européen "SoundObj" (The Sounding Object) [RBF03], qui propose une vue d'ensemble très exhaustive du domaine.

2.1.3 Properties of virtual sound sources. Spatial extent

La plupart des systèmes de rendu audio 3D temps-réel travaille dans l'hypothèse de sources sonores ponctuelles (donc d'ondes sonores sphériques). Comme nous le verrons par la suite, cette hypothèse permet un traitement simplifié des phénomènes de propagation. Comme une source sonore réelle, la source sonore est en générale caractérisée par sa réponse impulsionnelle et/ou une fonction de directivité qui encode l'atténuation du son suivant sa direction d'émission. Dans le cas le plus général, la source peut être décrite par une série de réponses impulsionnelles dépendant de la direction. Ces données doivent en général être mesurées sur la source, ce qui est très difficile à réaliser en pratique sur des sources naturelles et qui conduit à l'utilisation de modèles très simplifiés. L'hypothèse de source ponctuelle, infinitésimale, est particulièrement irréaliste. En pratique, aucun émetteur de son ne peut être considéré comme vraiment ponctuel. La solution à ce problème passe par l'échantillonnage des sources sonores étendues par un nuage de source sonores ponctuelles. Le problème principal réside alors dans la détermination du signal devant être émis par chacune de ces sources. Si un même enregistrement est utilisé, des effets d'interférence ("phasing" ou effet de filtre en peigne) apparaissent alors, liés aux retards de propagation entre les sources. Certaines approches proposent de créer des copies décorrélées [PB04c] d'un même enregistrement de manière à limiter ces problèmes. Mais elles restent toutefois très limitées. Une solution applicable dans certains cas est de réaliser des prises de son avec plusieurs microphones placés en champ proche des différentes parties émettrices de son de la source. Par exemple pour un véhicule, un microphone directif peut capturer les bruits du moteur, d'autres placés à proximité des roues vont capturer les bruits de contact pneumatique-surface [AWBW05]. Une autre approche, reposant sur le format d'encodage "Ambisonics" (voir Chapitre 3), réalise une décomposition de la directivité ou la réponse directionnelle d'une source sonore complexe sur une base d'harmoniques sphériques [Mal01, Men02]. Un autre type de prise de son peut être réalisée en utilisant une antenne de microphones. En combinant les multiples enregistrements obtenus, un microphone "virtuel" très directif peut être réalisé et orienté pour séparer les différents composantes d'une source sonore complexe [ME04a]. Néanmoins, ces solutions reposent sur des dispositifs de prise de son difficiles à mettre en uvre, peu accessibles en pratique et qui ne peuvent s'affranchir de problèmes liés aux transducteurs et aux traitements (calibration, rapport signal/bruit, bande passante réduite). Une alternative peut alors être la synthèse directe à partir de modèles physiques qui peut donner de bons résultats dans le cas de sources sonores étendues, comme c'est le cas dans les travaux [DYN04, DYN03] pour la simulation des phénomènes aerodynamiques ou du feu. La description et l'acquisition des propriétés des sources sonores, alors qu'elle est l'un des points clés d'un système de rendu audio en reste encore, à l'heure actuelle, l'un des aspects les moins développés.

2.2 Modeling sound propagation

Une fois le son émis par les sources il se propage dans l'environnement jusqu'à atteindre les oreilles de l'auditeur. Durant ce trajet, les ondes sonores interagissent avec l'environnement de multiples façons, en particulier en se réfiéchissant et diffusant sur les obstacles. Chaque réflexion, ou de manière plus générale trajet indirect du son, parvient à son tour aux oreilles de l'auditeur légèrement atténué et retardé dans le temps. L'ensemble de ces contributions ajoute au son émis par la source un effet car-

actéristique de l'environnement dans lequel elle se trouve ainsi que l'auditeur. Reproduire ces effets liés à l'environnement est primordial pour une bonne localisation 3D de la source sonore et une perception cohérente de l'environnement virtuel. Bien qu'il ne s'agisse plus de simuler une source sonore mais sa transformation, on retrouve ici la distinction entre la synthèse par échantillonnage, par modèle physique ou par analyse/synthèse. Ces approches se distinguent en pratique par le type de codage de la scène sonore, c'est-à-dire par les données d'entrée du processus de simulation : signal équivalent, caractéristiques acoustiques et géométriques ou descripteurs perceptifs. Cependant, elles aboutissent toutes à une représentation signal du filtrage par lequel le son brut émis par la source doit être transformé de manière à traduire les effets liés à la propagation du son dans l'environnement.

2.2.1 Acquiring impulse responses and rendering

Une première possibilité pour le rendu des effets environnementaux est d'acquérir directement des réponses impulsionnelles d'environnements réels. Historiquement, il s'agit de la première technique développée pour donner accès à l'écoute virtuelle d'un lieu réel ou d'une maquette (désigné en anglais par le terme "auralisation"). Elle constitue, pour la simulation de l'environnement, l'équivalent des techniques d'échantillonnage employées pour la synthèse des sources sonores. Le lieu dont on veut simuler les propriétés acoustiques est caractérisé par la mesure d'une réponse impulsionnelle qui consigne l'ensemble des transformations subies par le signal sonore entre la source et le récepteur. Cette réponse peut-être ensuite utilisée pour filtrer, par convolution, le signal émis par la source afin d'en simuler la présence dans le lieu considéré. Ne reposant pas sur un modèle mais sur l'enregistrement de la signature acoustique du lieu, cette technique offre a priori les garanties d'authenticité auditive de la simulation. Longtemps réservée au monde du laboratoire à des fins d'évaluation perceptive de la qualité acoustique d'environnements tels qu'une salle de concert [JAS83, AS83] ou un habitacle de voiture [GK96, FU97], elle est entrée plus récemment dans le domaine de la production audio. Certains réverbérateurs artificiels, utilisés dans les studios d'enregistrement et de post-production, proposent un vaste répertoire de réponses impulsionnelles enregistrées dans différentes salles prestigieuses (Concertgebouw d'Amsterdam, la Musikverein de Vienne, la Scala de Milan, etc), ou encore dans des lieux génériques (une gare, une forêt, une salle de bain, une grotte, etc). L'authenticité du résultat doit être cependant nuancée en observant les conditions d'obtention de ces réponses. Notamment, il convient de considérer les limites du système mesuré qui englobent non seulement les informations acoustiques liées à la salle mais également celles du haut-parleur et du dispositif microphonique servant à la mesure, en particulier leurs caractéristiques de rayonnement. Ainsi la convolution d'un signal de violon par la réponse impulsionnelle d'une salle de concert simule en fait la diffusion de ce signal de violon par le haut-parleur utilisé lors de la mesure et non la présence du violon lui-même dans la salle. Cette approche souffre également des limites inhérentes au caractère figé de la situation enregistrée. Notamment, elle interdit a priori toute interactivité puisque la réponse est établie pour une position et une orientation déterminées de la source et du récepteur. Une solution consiste à opérer un échantillonnage spatial de l'environnement en multipliant les acquisitions pour diverses positions et/ou orientations du récepteur entre lesquelles on effectue une interpolation lors de la convolution de manière à simuler la déambulation de l'auditeur dans l'environnement [Pel99, HKP$^+$99]. Pour limiter les cots de calcul et d'emplacement mémoire liés à l'interpolation entre plusieurs réponses, on restreint généralement cette opération à la partie précoce de la réponse (typiquement la première centaine de millisecondes) en observant que les caractéristiques de la réverbération tardive varient peu en fonction de la position de la source ou du récepteur dans la salle. A l'évidence, cette approche ne peut être cependant extrapolée au cas de sources et de récepteurs simultanément en mouvement car elle conduirait à une combinatoire de points de mesure trop lourde. Par ailleurs, le recours à l'interpolation pour simuler la transition entre deux points

de mesures voisins dans une même salle est très éloigné de la réalité physique et peut s'accompagner d'artefacts audibles (effet de filtrage en peigne). Les réponses impulsionnelles consignent également les caractéristiques du système de prise de son utilisé pendant l'acquisition et se trouvent par conséquent associées à un format et un système de diffusion déterminés. Ainsi, une base de réponses impulsionnelles de salle mesurées avec une tête artificielle ou un couple stéréo condamne respectivement à une écoute des simulations sur un casque ou sur un couple d'enceintes par exemple [JWL98]. Cette contrainte peut être cependant levée en ayant recours à des antennes microphoniques [LBM03, HBd01] qui encodent de manière générique le champ sonore au voisinage du point de mesure (voir également Chapitre 3). Cet encodage multi-canal permet alors de s'adresser à toute une gamme de dispositifs d'écoute et rétablit même un certain degré d'interactivité en autorisant, par exemple, la rotation virtuelle du point d'écoute. Symétriquement, la caractérisation de la réponse de la salle par un ensemble de sources de directivité élémentaire permet de simuler a posteriori la réponse de la salle pour des sources de directivité complexe ou pour différentes orientations [WM04]. En résumé, l'exploitation d'une base de données de réponses impulsionnelles de salles enregistrées permet de simuler de manière simple la présence d'une source dans un espace acoustique réel avec une qualité de rendu sonore optimale. Reposant, par définition, sur une étape préalable d'acquisition, elle est réservée à la simulation de situations statiques ou faiblement interactives n'exigeant pas de mise en cohérence avec la représentation visuelle d'un espace. La qualité du rendu sonore lui confère cependant un statut d'étalon pour juger de la pertinence des algorithmes de synthèse d'effet de salle et des approches basées sur une modélisation perceptive ou physique.

2.2.2 Physically-based models for sound propagation

Une première approche pour reproduire les effets de propagation du son dans un environnement totalement virtuel est de les simuler à partir de modèles physiques de la propagation du son. Deux grandes familles d'approches peuvent être distinguées : les modèles ondulatoires qui prennent en compte l'ensemble des effets caractéristiques de la propagation des ondes sonores tels que les interférences, la diffraction par les obstacles, de manière unifiée. Ces approches utilisent des techniques numériques pour déterminer directement une solution à l'équation de propagation des ondes dans l'environnement voulu. Une deuxième catégorie d'approches dérive d'approximations des techniques précédentes dans le cas des haute-fréquences. Elles utilisent des calculs géométriques pour déterminer la propagation de rayons sonores depuis les sources jusqu'à l'auditeur. Très proches des techniques utilisées en synthèse d'image pour les calculs d'éclairage global (lancer de rayon par exemple), elles ne traitent pas de manière unifiée des phénomènes tels que la diffraction, qui nécessite des extensions supplémentaires (comme la Théorie Géométrique de la Diffraction). Elles demeurent néanmoins pour l'instant les méthodes les plus efficaces et les plus utilisées dans le domaine des simulations acoustiques interactives.

Modèles ondulatoires. Simulations par éléments finis

Les techniques par éléments finis résolvent numériquement l'équation d'onde (et les conditions aux limites associées) à une fréquence donnée en subdivisant l'espace en éléments. On notera ici la différence entre ces techniques d'éléments finis acoustiques s'appliquant à l'équation d'onde et les techniques d'éléments finis mécaniques évoquées dans la section 1.3 pour la synthèse à base physique. L'équation d'onde est alors exprimée comme un nombre fini d'équations linéaires pour chaque élément. La forme intégrale de frontière de l'équation d'onde, i.e., l'équation de Green ou Helmholtz-Kirchhoff peut, elle, être résolue en ne subdivisant que les surfaces de l'environnement et considérant que la pression (ou la vitesse particulaire) est une combinaison linéaire d'un nombre fini de fonctions de bases sur les éléments. Les coefficients de cette combinaison linéaire sont alors les inconnues d'un système linéaire qui doit être

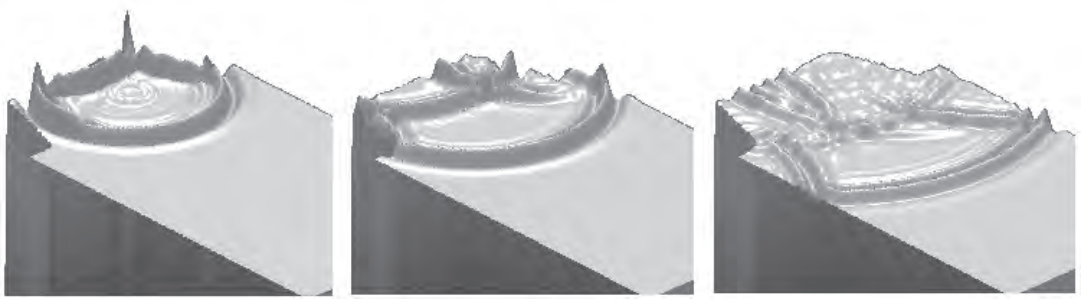


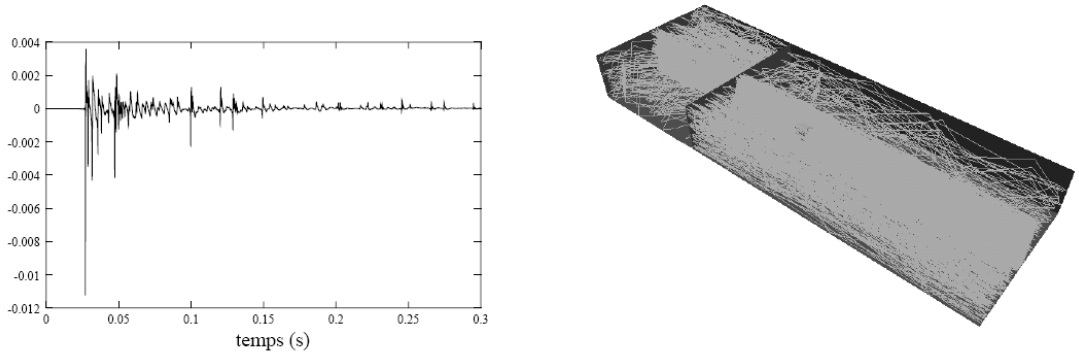
Figure 2.2: Visualisation de la propagation d'une onde sonore en 2D dans une salle calculée par une méthode de différence finie temporelle. Image reproduite d'après [SV01].

résolu. Pour cela on peut imposer que les équations soient satisfaites pour une ensemble de points nodaux (méthodes dites de collocation) ou utiliser un critère de convergence global (méthodes dites de Galerkin). Ce critère exprime que la solution au système doit être une fonction de l'espace engendré par les fonctions de base la plus proche de la véritable solution. Les deux méthodes sont en général comparables en complexité. Les techniques par éléments finis permettent d'obtenir une solution précise à l'équation d'onde. Néanmoins, dans le cadre d'applications interactives, elles ne peuvent que se limiter à des basses fréquences ou environnements très simples puisque leur complexité (temps de calcul et mémoire) augmente de manière très importante avec la fréquence. De plus, afin de reconstruire une réponse impulsionnelle, un grand nombre de simulations est nécessaire puisqu'il faut reconstruire la fonction de transfert pour pouvoir l'inverser (par transformée de Fourier inverse). De manière duale aux approches précédentes qui travaillent en espace de fréquence, les techniques par différences finies, comme le "digital waveguide mesh", permettent de simuler la propagation du son dans le domaine temporel [DS96, SRT94, SV01] (voir Figure 2.2).

Modèles géométriques : Rayons sonores et échanges radiatifs

Les techniques d'acoustique géométrique modélisent les effets de l'environnement en se basant sur la théorie des rayons sonores, suivant laquelle le son se propage le long de trajectoires rectilignes dans l'espace. Elles reposent sur l'hypothèse que la longueur d'onde des ondes sonores considérées est beaucoup plus petite que la taille des surfaces ou obstacles dans l'environnement. En conséquence, elles constituent des solutions haute-fréquence au problème de la propagation acoustique. En pratique, elles donnent lieu à des algorithmes géométriques très proches de ceux utilisés en synthèse d'image pour la propagation lumineuse. Ces algorithmes permettent de construire l'ensemble des chemins ou rayons sonores le long desquels le son peut se propager depuis un point source jusqu'à un point récepteur . Les propriétés géométriques du chemin, comme sa longueur, et des modèles mathématiques permettent alors de construire un filtre approximant les effets de propagation (i.e., directivité de la source et du récepteur, diffusion atmosphérique, réfiectance des surfaces, coefficients de diffraction des arêtes, etc.) le long de chaque chemin. La réponse impulsionnelle est finalement obtenue en sommant les filtres de tous les chemins (Figure 2.3).

Le problème principal dans le cadre d'applications interactives est de déterminer de manière efficace les chemins de propagation possibles entre la source et le récepteur et les différents évènements associés (i.e., reflexion, diffraction, etc.). Le milieu étant en général considéré homogène, ces chemins sont donc linéaires par morceaux et leurs sommets vont se situer sur les surfaces (réflexion) ou les arêtes (diffraction) des obstacles dans l'environnement. Un point clé des simulations d'acoustique géométriques, qui



(a) Réponse impulsionnelle (b) Chemins de propagation

Figure 2.3: Réponse impulsionelle (a) correspondant à 353 chemins de propagation (b). Ces chemins sont calculés pour des séquences d'au maximum 10 réflexions successives sur les parois entre une source et un récepteur ponctuels. Les deux pièces sont reliées par une porte.

les distinguent de leurs homologues en synthèse d'image est l'utilisation privilégiée de modèles de reflexions spéculaires sur les surfaces. Cette hypothèse est justifiée par le fait que pour les longueurs d'onde et les tailles des obstacles considérées (e.g., les murs d'un batiment), les surfaces peuvent être assimilées à des miroirs. Les modèles géométriques de la diffraction [Kel62, KP74, Han81, MPM90], qui considèrent les arêtes de l'environnement comme des sources de rayons diffractés, reposent également sur cette même hypothèse "spéculaire". En conséquence du principe de Fermat, entre une source et un récepteur ponctuels, une séquence de réflexion sur des surfaces et diffraction sur des arêtes va donner lieu à un chemin de propagation unique qui est le plus court chemin intersectant les surfaces et les arêtes. A chaque intersection, les angles d'incidence et de reflexion (resp. diffraction) avec la normale à la surface (resp. la direction de l'arête) sont égaux. L'existence d'un unique chemin pour une séquence réflexion et diffraction permet de représenter la contribution d'un trajet indirect du son comme l'apport d'une source sonore virtuelle secondaire appelée source-image.

Ce concept de source-image est à la base d'une technique fondamentale utilisée pour simuler des réflexions acoustiques dans des environnements virtuels [AB79, Bor84, KOK93, Hei93, KKF93]. Cette technique détermine donc les séquences de réflexions spéculaires dans l'environnement en construisant de manière récursive des sources virtuelles qui sont les images-mirroir successives de la source sonore principale par chaque surface de l'environnement (voir Figure 2.4). Le principal avantage de cette technique est sa robustesse. Elle permet de garantir que tous les chemins spéculaires jusqu'à un ordre de réflexion donné sont trouvés. Néamoins, elle ne permet de simuler que des réflexions spéculaires et leur complexité croit exponentiellement en O(nr) où n est le nombre de surfaces et r l'ordre de réflexion. De plus dans tous les environnements (sauf les plus simples comme un parallélépipède), des tests de visibilité/validité doivent être effectués pour chacune de ces sources-images puisqu'elle peuvent ne pas correspondre à un chemin réalisable à cause d'un obstacle, ou de l'étendue finie des surfaces (Figure 2.4(c)). Dans le cas d'environnements parallélépipédiques simples, ces techniques sont néanmoins très efficaces puisque les sources images correspondant à différentes permutations de réflexions sur les surfaces se retrouvent superposées en des positions identiques qui s'alignent sur une grille. De plus pour chaque point récepteur dans l'environnement, une seule et unique permutation de reflexions donnant lieu à la même source image est visible. Certains systèmes simplifient donc les environnements complexes en utilisant leur boîte englobante pour effectuer les calculs de source-image. D'autres techniques permettent néanmoins de construire l'ensemble des chemins de propagation valides de manière plus efficace.

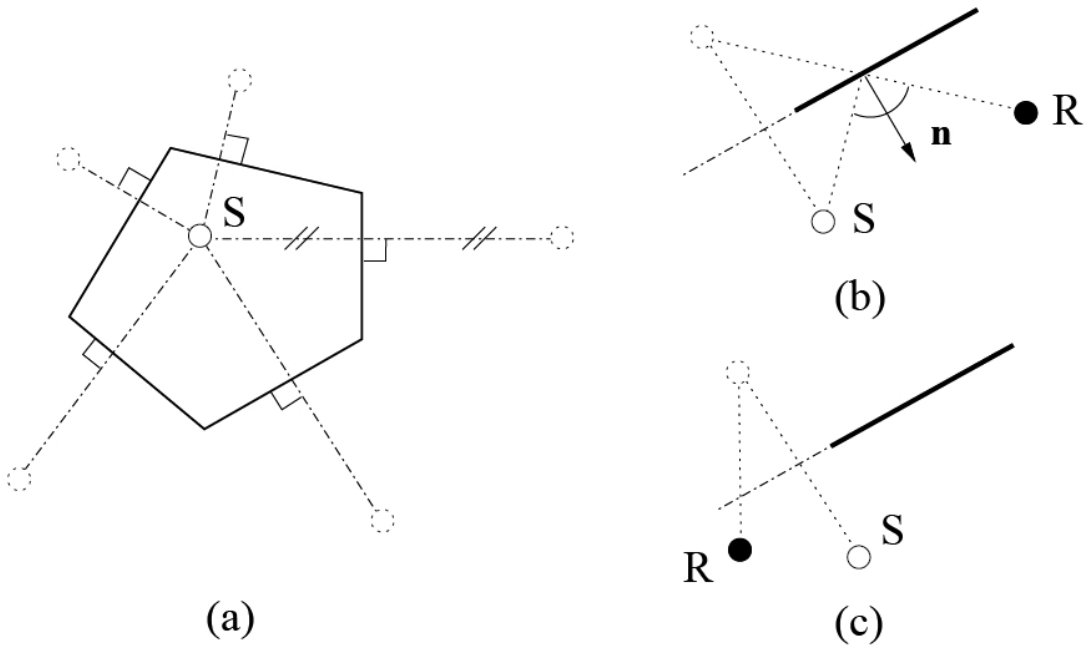


Figure 2.4: Méthode des sources-images en 2D. La Figure (a) montre une source sonore (S) et ses sources images de premier ordre pour un contour pentagonal. La Figure (b) représente une source image valide pour une position de récepteur (R) ; la Figure (c) représente une configuration invalide car le chemin réfléchi entre la source virtuelle et le récepteur n'intersecte pas le réflecteur.

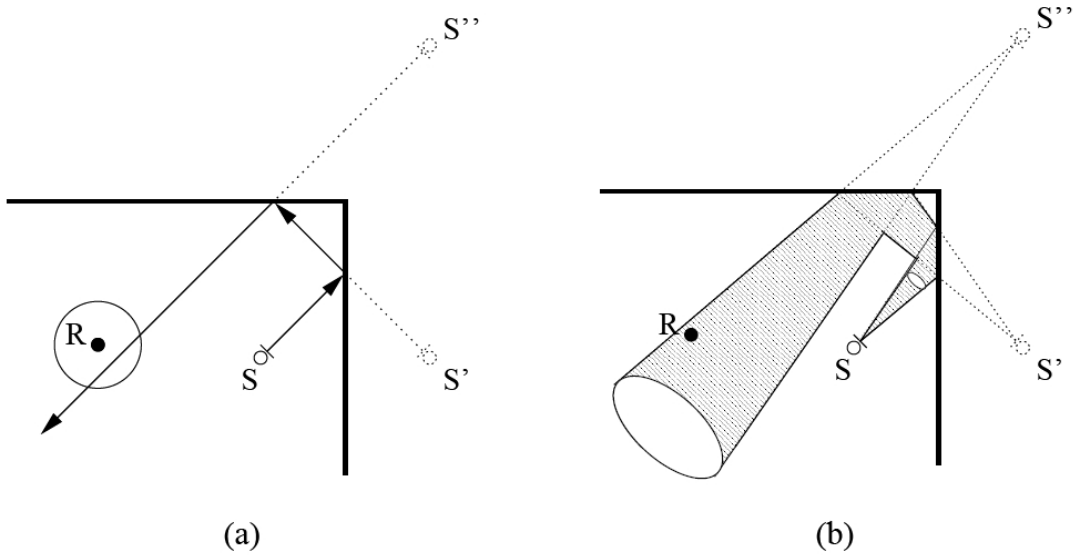


Figure 2.5: (a) Calcul des sources-images par lancer de rayons : Des rayons sont tirés depuis la source et réfléchis spéculairement jusqu'à ce qu'ils atteignent un volume de réception où leurs contributions sont comptabilisées. Les rayons sont ensuite prolongés jusqu'à ce que leur énergie soit trop faible. (b) Par lancer de cônes : des cônes sont tirés depuis la source et réfléchis jusqu'à qu'ils contiennent le récepteur.

Les approches dites de lancer de rayon construisent les chemins de propagation entre une source et un récepteur en générant des rayons depuis la position de la source (Figure 2.5 (a)) et en les suivant à travers l'environnement jusqu'à ce qu'un ensemble de chemins appropriés atteigne une représentation de la position de l'auditeur (une sphère par exemple) [Leh93, Dal96, Eme95, vMM93, Nay93]. Là encore, ces méthodes sont simples à mettre en uvre et ne dépendent que de calculs d'intersection rayon/surface dont la complexité est sub-linéaire du nombre de surfaces. Elles sont également générales et permettent de traiter des surfaces dont la réflectance est plus complexe (par exemple par des techniques d'échantillonnage de Monte Carlo) ainsi que des surfaces courbes. Toutefois, un inconvénient majeur de telles techniques est l'aliassage spatial [Leh93] puisque l'espace continu des rayons 3D n'est échantilloné que par un nombre limité de rayons, ce qui conduit à des erreurs dans la solution obtenue. Par exemple, la position du récepteur ou les arêtes de diffraction sont souvent approchées par des volumes (pour admettre une intersection avec un rayon) ce qui peut conduire à des fausses intersections et des chemins pris en compte de manière multiple. A l'opposé des chemins de propagation importants peuvent être manqués par tous les échantillons. Afin de minimiser les risques d'erreur, les systèmes de lancer de rayons utilisent souvent de nombreux échantillons au détriment du temps de calcul. Enfin, les résultats dépendent de la position des sources et de l'auditeur, ce qui rend de telles techniques peu adaptées à des applications interactives où ceux-ci se déplacent.

En réponse aux problèmes d'aliassage des techniques de lancer de rayon ont également été développées des techniques dites de lancer de faisceaux [HH84, FCE$^+$98, FMC99, MF00, LCM07]. Celles-ci classifient l'espace des rayons émis depuis la source en traçant récursivement des faisceaux (i.e., des groupes de rayons) pyramidaux à travers l'environnement (voir Figure 2.7). Pour chaque faisceau, on teste les intersections potentielles avec les polygones de l'environnement du plus près au plus lointain (de manière à ne pas considérer de polygone avant que tout ceux qui le cachent au moins partiellement aient été considérés). A chaque fois qu'une intersection faisceau/polygone est détectée, le faisceau est fenêtré afin de

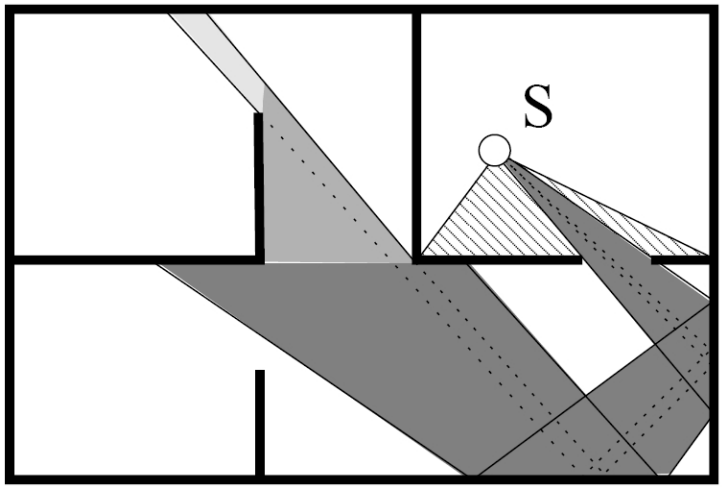


Figure 2.6: Réflexion et fenêtrage des faisceaux générés depuis la source (S).

lui retirer la zone d'ombre créé par le polygone (Figure 2.6). Un faisceau transmis correspondant à cette zone d'ombre est créé ainsi qu'un faisceau réfiéchi en construisant le faisceau miroir du faisceau transmis par rapport au plan du polygone. D'autres faisceaux, correspondant à d'autres types d'évènements comme la diffraction par les arêtes du modèle, peuvent être également créés au besoin. Ces approches ont déjà été utilisées pour une variété d'applications, comme l'acoustique [FCE$^+$98, FMC99, MF00, MOD96, SK95, Lew93], les calculs d'éclairage global [CC95, Fuj88, GH98, Hai91, HH84, Wat90], la propagation d'ondes radio [KGW$^+$99, For96] et la détermination de visibilité [Jon71, Tel92, LG95, FST92].

Les approches de lancer de faisceaux permettent d'énumérer de manière beaucoup plus efficace l'ensemble des sources-images valides dans des environnements généraux. En effet, un faisceau représente la région de l'espace pour laquelle la source-image correspondant (le sommet du faisceau) est visible. Les sources-images d'ordre supérieur ne doivent donc être considérées que pour les polygones intersectant le faisceau, ce qui permet de réduire l'arbre de recursivite de manière significative. L'avantage des techniques de lancer de faisceaux sur le lancer de rayon est qu'elles tirent parti de la cohérence spatiale, l'intersection entre un faisceau et une surface représentant une infinité d'intersections rayon/surface. Le lancer de faisceau polyhédrique ne souffre donc pas des problèmes d'aliassage spatial du lancer de rayon, ni des problèmes de recouvrement du lancer de cône [Ama84, MvMV93] ((Figure 2.5 (b)), puisque l'ensemble 2D des directions dans lesquelles le son peut se propager à partir de la source est couvert exactement. Ces méthodes peuvent donc énumérer tous les chemins de propagation potentiels jusqu'à un certain critère de terminaison (l'ordre, la longueur, etc.) sans en oublier. Elles permettent également l'intersection sans aliassage avec les arêtes pour le traitement unifié de la diffraction [TFNC01]. Enfin, il est possible de combiner tracé de faisceaux depuis les sources et le récepteur pour trouver les chemins de propagation de manière plus efficace [FMC99]. Dans le cadre d'applications interactives, ces approches permettent également de précalculer les faisceaux et les stocker dans une structure de données appropriée (un arbre) pour permettre une reconstruction temps-réel des chemins de propagation [FCE$^+$98]. Par exemple, des faisceaux pré-calculés pour des positions de sources fixées permettent de recalculer les chemins de propagation interactivement lorsqu'un auditeur se déplace dans l'environnement. Ou bien, des faisceaux calculés pour des régions sources étendues peuvent être mis à jour de manière asynchrone afin de permettre la génération des chemins de propagation entre sources et récepteurs mobiles en temps interactif [FMC99] (voir Figure 2.7).

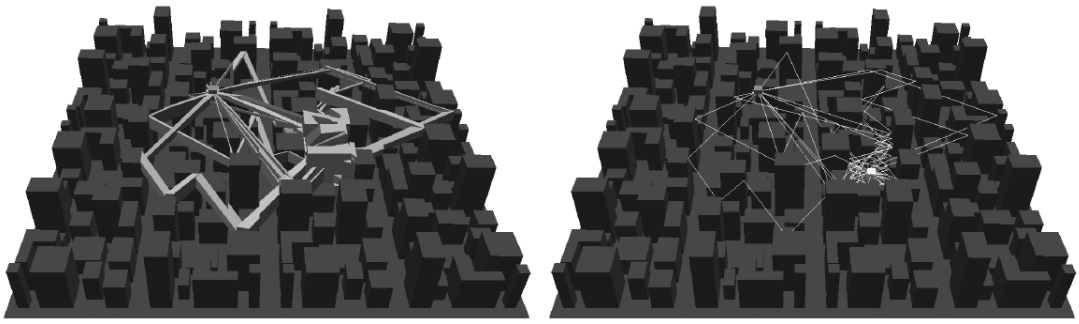


Figure 2.7: La structure des faisceaux (à gauche) peut être précalculée dans le cas de sources fixes et interrogée en temps-réel pour mettre à jour les chemins de propagation jusqu'à un auditeur mobile. (à droite). D'après [FCE+98].

La principale difficulté des approches de lancer de faisceau est la complexité des opérations géométriques nécessaires à l'intersection et au fenêtrage ("clipping") des faisceaux avec les polygones de l'environnement. En outre, des problèmes de robustesse liés aux erreurs numériques peuvent conduire à la construction de faisceaux dégénérés. Néanmoins, plusieurs méthodes ont été proposées pour accélérer les calculs géométriques nécessaires à base de subdivisions spatiales de type arbre BSP [DKW82, DKW85], graphes d'adjacences de cellules [Jon71, FCE+98, Tel92, LG95, FMC99], couches de triangulations 2D [For99] et approximations par axe médian [KUG93]. Ces méthodes sont particulièrement efficaces dans le cas d'environnements simples ou dans lesquels peu de surfaces sont visibles simultanément (e.g., les villes ou des environnements intérieurs). Le lancer de faisceau est également difficile dans le cas de scènes avec des surfaces courbes. Néanmois, les approches utilisant des faisceaux simplifiés de manière prudente (en général en remplacant le faisceau exact par un polyhèdre englobant plus simple) combinés avec une post-validation des chemins [FCE+98, FMC99, TFNC01] sont bien adaptées à ce problème. Des techniques par éléments finis ont également été utilisées pour simuler des transferts énergétiques diffus (i.e., lambertiens) entre les surfaces de manière similaire aux approches de radiosité en synthèse d'image [CW93b, SP94]. Ces techniques permettent de modéliser de manière efficace les caractéristiques de décroissance temporelle de l'énergie dans la réverbération tardive ou des réflexions diffuses [Lew93, SK95, Dal96]. Néanmoins, elles ne permettent pas une reconstruction exacte de la réponse impulsionnelle puisqu'elles restent purement énergétiques. Des approches proposant des représentations statistiques de la phase du signal (par exemple, une phase aléatoire) permettent toutefois de regénérer une réponse impulsionnelle sous la forme d'un bruit coloré dont la décroissance (pour différentes bandes de fréquences par exemple) est fournie par la solution de "radiosité acoustique" [Lew93].

Diffraction, occultation et transmission

Les techniques géométriques basées sur la théorie des rayons sonores ont également été étendues pour prendre en compte la diffraction des ondes sonores sur les obstacles. Une des approches se base sur la Théorie Géométrique de la Diffraction (TGD), introduite par Keller [Kel62, KP74, Han81, MPM90], qui considère les arêtes (discontinuités de surface) du modèle comme sources de rayons secondaires diffractés (voir Figure 2.8 et 2.9).

Bien que la construction de chemins de propagation comprenant des diffractions est plus complexe, les techniques de lancer de faisceau présentées précédemment ont en particulier été étendues pour pren-

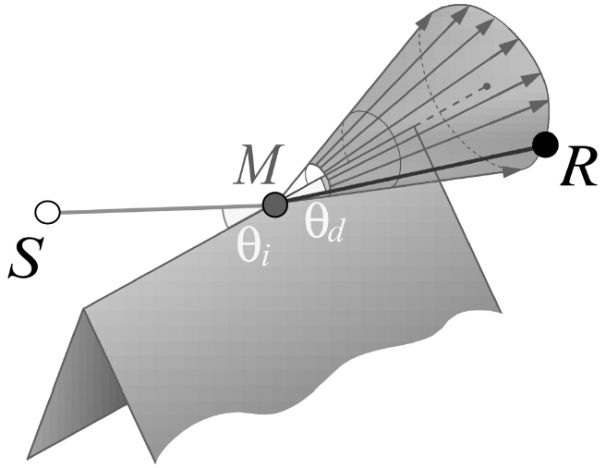


Figure 2.8: Selon la théorie uniforme de la diffraction, un rayon incident ρ sur une arête donne naissance à un cône de rayons diffractés. L'angle d'ouverture fid du cône est égal à l'angle θ_i entre le rayon incident et l'arête (i.e., l'axe du cône). Pour une position donnée du recepteur, un rayon unique décrit le champ diffracté.

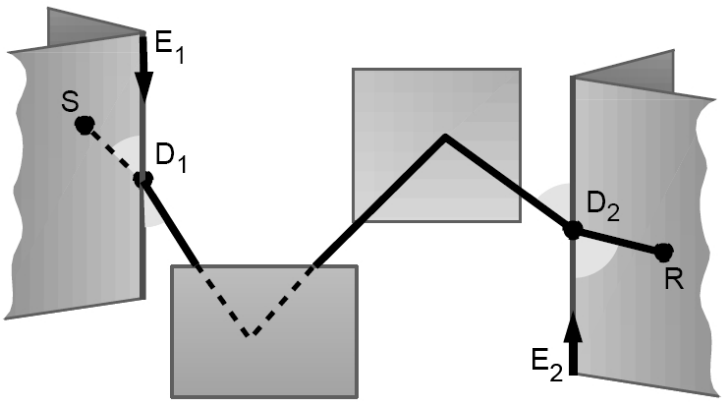
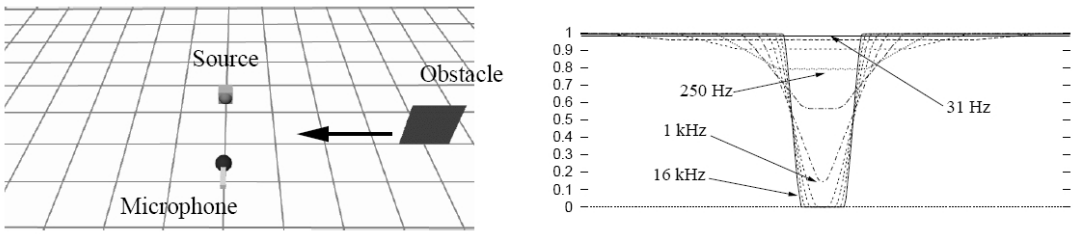


Figure 2.9: Un chemin de propagation du son comprenant une diffraction, deux réflexions spéculaires et une seconde diffraction. Suivant la théorie géométrique de la diffraction, les deux points de diffraction D_i sont déterminés par des contraintes d'égalité angulaire aux arêtes correspondantes E_i.

(a) (b) (c)

Figure 2.10: Utilisation du rendu 3D câblé pour le calcul de la ”visibilité sonore”. (a) Vue 3D montrant un microphone, une source, les premiers ellipsodes de Fresnel associés (pour des fréquences de 400 et 4000 Hz) ainsi que des obstacles. (b) Visibilité depuis la source à 400 Hz. (c) Visibilité depuis la source à 4000 Hz. La zone blanche circulaire correspond à la première zone de Fresnel à mi-distance de la source et du récepteur.



Environnement de test «Visibilité sonore » au cours du temps

Figure 2.11: volution de la ”visibilité sonore ” en fonction du temps lorsqu'un obstacle en mouvement passe entre la source et le récepteur, estimée par occultation des premiers ellipsoides de Fresnel. L'obstacle est une plaque carrée de surface égale à $1m^2$. La source et le récepteur sont distants de 2 m. Les valeurs sont données en bandes d'octaves de 31 Hz à 16 KHz.

dre en compte ce phénomène de manière efficace dans le cadre d'applications temps-réel [TFNC01]. Malgré cela, simuler la diffraction des ondes sonores reste un problème difficile et coteux. Dans la plupart des applications, les obstacles sont donc considérés comme de simples ”écrans” acoustiques qui atténuent le son. Des modèles approximatifs, utilisant par exemple des calculs d'occultation des ellipsoides de Fresnel [TG97] ont été proposés et permettent un traitement qualitatif plus efficace des phénomènes d'occultation et transmission sonore dans le cadre d'obstacles complexes, tout en restant réalistes (voir Figures 2.10 et 2.11).

2.2.3 Generic models for environmental effects

Les travaux sur la caractérisation perceptive de l'effet de salle ainsi que les considérations physiques sur la propagation acoustique dans les salles invitent à décrire la réponse impulsionnelle associée à la situation d'écoute de l'auditeur sous forme d'un modèle simplifié. Ce modèle est basé sur une représentation temps-fréquence caractérisée notamment par différentes sections de support temporel

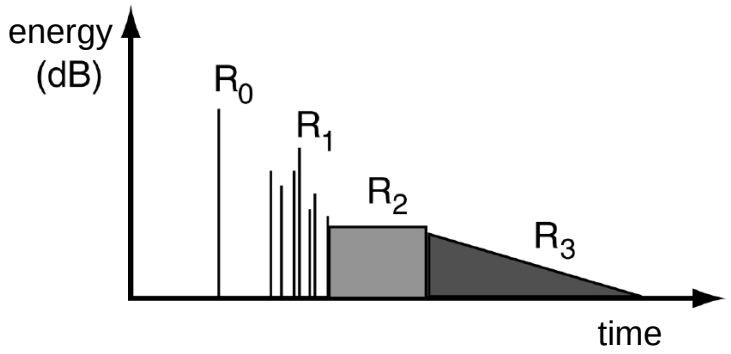


Figure 2.12: Représentation schématique d'un modèle générique d'effet de salle.

croissant. Sans que le modèle présenté (Figure reffig:roomeffect) soit normatif, il permet d'illustrer les principes généralement retenus. Une première section R0 est dédiée exclusivement au son direct, compte tenu de l'importance de son contrôle pour la perception de la localisation de la source, de ses caractéristiques spectrales, etc... La section R1 est constituée d'un jeu restreint de réflexions précoces contenues typiquement dans l'intervalle]0;40ms] et que l'on désire pouvoir régler de manière individuelle. Du point de vue perceptif, bien que ces premières réflexions soient subjectivement intégrées à la perception du son direct, leur distribution temporelle et spatiale peut en effet modifier le timbre ainsi que la localisation et la largeur apparentes de la source. Plus on s'éloigne dans le temps, plus la distribution spatiale et temporelle des réflexions répond à un comportement statistique. Cette propriété rend à la fois impossible et inutile, du point de vue perceptif, une description individuelle des différentes réflexions et suggère une description davantage morphologique. Pour la partie tardive, on a donc recours à la description de l'enveloppe de décroissance et à différents paramètres relatifs à sa structure interne, comme la densité temporelle de réflexions ou la densité modale associée. Dans la Figure reffig:roomeffect, la section R3, représente la queue de réverbération obéissant à une décroissance exponentielle caractérisée classiquement par le temps de réverbération. La prise en compte de processus d'intégration conditionnelle des réflexions tardives conduit à distinguer la plage de réflexions R2 typiquement située dans l'intervalle [40;100] ms.

En parallèle de cette description temporelle, il faut également considérer la dépendance fréquentielle et spatiale. Le choix du découpage fréquentiel répond à un compromis entre la finesse de contrôle souhaitée et la puissance de calcul requise. De même que pour la dimension temporelle, les propriétés perceptives et physiques suggèrent d'adopter une description spatiale dont la résolution varie au cours du temps. L'importance du phénomène de précédence, qui désigne les conditions de fusion des premières réflexions et du son direct et la dominance de ce dernier sur la localisation de la source, impose une description spatiale fine de l'effet de salle précoce. En revanche, la résolution spatiale peut être relâchée pour les sections plus tardives [Pel01b] pour laisser place, là encore, à des descripteurs de nature statistique comme le degré de corrélation interaurale. Dans les situations interactives reposant sur un rafraîchissement régulier des paramètres en fonction des commandes de l'utilisateur, ces mêmes considérations autorisent de relâcher les contraintes de latence et de taux de rafraîchissement pour les sections tardives de l'effet de salle.

L'intérêt d'un algorithme paramétrique d'effet de salle tel qu'exposé dans le paragraphe précédent est multiple. Outre le gain apporté en termes de cot de calcul par rapport à une convolution, il permet une grande flexibilité de contrôle et d'adaptation par rapport au contexte de diffusion. Notamment, il ne passe pas par l'intermédiaire d'une description sous forme d'une réponse impulsionnelle qui présente

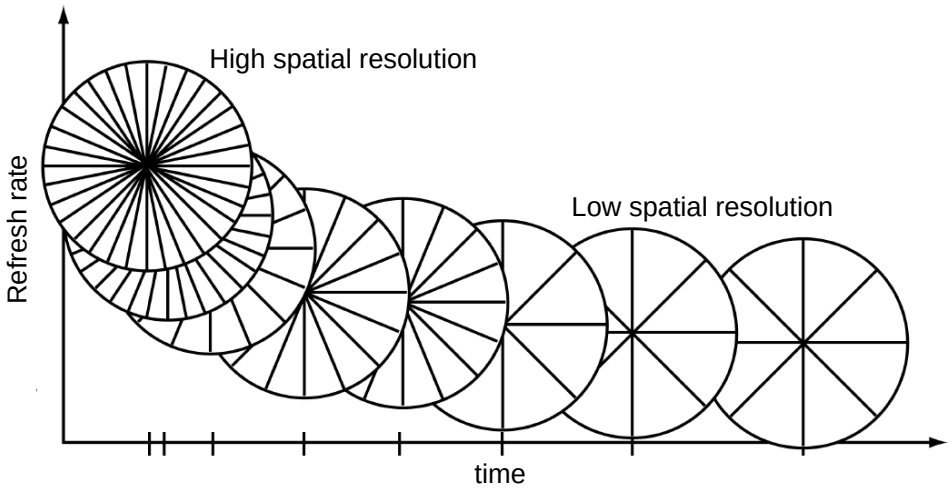


Figure 2.13: Evolution de la résolution spatiale et de la fréquence de rafraichissement d'un modèle générique d'effet de salle [Pel01b].

l'inconvénient d'encoder simultanément le dispositif de captation, comme évoqué au paragraphe 1.4.1. La séparation des étapes de synthèse de la distribution temporelle et de la distribution spatiale permet de s'adresser aussi bien à des dispositifs individuels comme le casque qu'à dispositifs destinés à des écoutes collectives comme les systèmes surround, ambisonique ou holophonique (cf. Chapitre 3). L'intérêt du modèle générique et de son algorithme associé est de pouvoir spécifier l'effet de la salle sans référence à une description géométrique donnée. On comprend l'intérêt d'une telle approche pour les applications audio et notamment à vocation musicale dans la mesure ou l'effet recherché s'adresse en priorité, voire exclusivement, à la perception auditive.

Par extension ce modèle est également adapté aux situations ne nécessitant pas la reconstruction exacte des propriétés acoustiques d'une salle donnée ou d'une scène décrite par ses caractéristiques architecturales. Compte tenu des propriétés de la perception, il est licite dans la plupart des cas de relâcher le critère d'authenticité de la simulation pour un critère moins contraignant de plausibilité du résultat perceptif. Pour les situations de réalité virtuelle courantes telles que les jeux ou les applications ne nécessitant pas un fort degré de cohérence entre le rendu sonore et visuel le modèle générique présenté ci-dessus peut être associé à un modèle physique statistique. Sans recourir à une description exhaustive des parois de la salle, ce modèle permet de traduire la dépendance de l'enveloppe temps-fréquence en fonction de données architecturales globales comme le volume et absorption totale de la salle, et en fonction de la distance entre la source et le récepteur [Jot97, Jot99]. Lorsque qu'un niveau supérieur de cohérence visuo-auditive est requis, le contrôle individuel des premières réflexions peut-être asservi aux informations fournies par des algorithmes de simulation de la propagation physique dans les salles tels que ceux décrits dans la Section 2.2.2. Cependant, ce modèle est effectivement limité aux situations répondant aux propriétés stochastiques du champ réverbéré dans les espaces clos. Par ailleurs, la plupart des attributs perceptifs utilisés pour décrire la qualité acoustique des salles, est issue de recherches effectuées dans le cadre restreint des salles à vocation musicale comme les salles de concert ou d'opéra ce qui en limite la portée et la pertinence pour des situations couramment rencontrées dans les applications de réalité virtuelle comme, par exemple, les lieux extérieurs (contexte urbain, forêt,...) qui s'accompagne pourtant d'effets acoustiques notables. Dans ces cas, seul le recours à une modélisation physique exhaustive peut garantir une simulation réaliste.

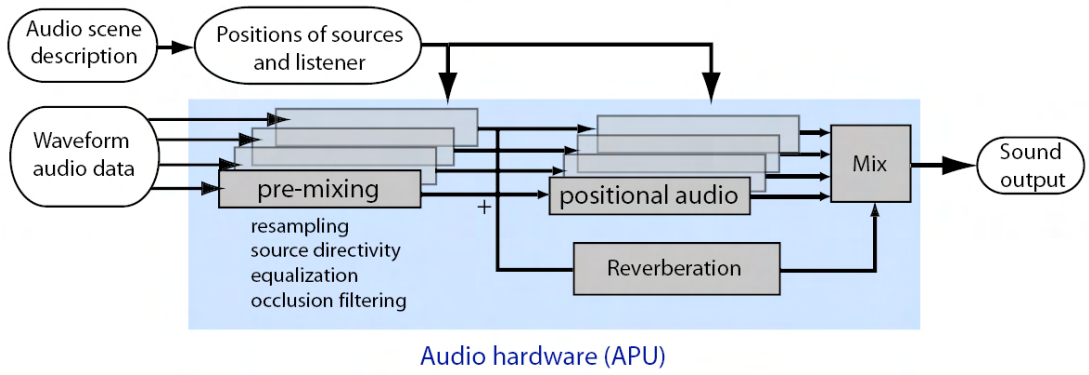


Figure 2.14: Vue d'ensemble d'un pipeline de rendu sonore spatialisé. La partie encadrée correspond à l'implémentation hardware disponible sur la plupart des systèmes dédiés et cartes-son grand public.

2.2.4 Integrating propagation effects in the rendering pipeline

D'une manière générale, tout effet appliqué au son se traduit dans le pipeline de rendu sonore par un filtrage numérique. Néanmoins, l'implémentation de ce filtrage peut être très différente suivant les performances désirées et les contraintes de l'application. La Figure 2.14 propose une vue d'ensemble d'un pipeline de rendu sonore classique dans les applications de réalité virtuelle dont nous détaillons à présents quelques points caractéristiques.

Rendering direct sound and early reflections

En règle générale, le son direct et les chemins indirects précoces (i.e., les sources-images) subissent les même traitements : ils sont d'abord ré-échantillonnés pour prendre en compte le retard de propagation de la source à l'auditeur. Cette "ligne à retard" peut être réalisée grâce à divers procédés d'interpolation dont le compromis entre qualité et rapidité peut être ajusté suivant les besoins ou contraintes de l'application [ZB94, Zöl02]. Ce ré-échantillonage permet également de prendre en compte et restituer l'effet Doppler dans le cas de sources mobiles. Cet effet est un indice perceptif majeur pour déterminer la vitesse et l'approche/éloignement d'une source. Ensuite, les signaux sont filtrés afin de prendre en compte les effets de propagation (atténuation atmosphérique, occultation, etc.) et les filtres spécifiques liés au dispositif de spatialisation utilisé (par exemple des fonctions de transfert des oreilles de l'auditeur, cf. autre article). Pour des raisons d'efficacité, des filtres FIR courts ou IIR d'ordre faible sont en général utilisés. Il faut noter ici que le filtrage lié à la source doit être appliqué avant la prise en compte de l'effet Doppler. Dans beaucoup de cas, il faut également noter que la prise en compte de l'effet Doppler n'est pas effectuée grâce à une ligne à retard mais simplement en compressant ou dilatant le signal (ou de manière duale en transposant ses composantes fréquentielles). Cette méthode ne permet donc pas de restituer les interférences qui peuvent exister entre différents chemins de propagation. Les paramètres de filtrage ainsi que les délais de propagation associés aux différents chemins doivent être recalculés dans le cas où l'auditeur et/ou les sources sonores se déplacent. En règle générale, les pipelines de rendu audio travaillent sur des trames de signaux source, dont la durée est d'environ 20ms (entre 30 et 60Hz). Les paramètres des filtres sont alors recalculés à chaque trame et éventuellement interpolés sur la durée d'une trame de traitement. Suivant les spécificités et contraintes de l'application, ces traitements peuvent être implémentés de manières très différentes et plus ou moins simplifiées [Mil01, MW02, EAX04, SHLV99]. En l'absence de standard équivalent, il est donc difficile de définir plus en détail un pipeline de traitement que l'on pourrait mettre en parallèle avec le pipe-line

de rendu des cartes graphiques par exemple.

Rendering reverberation effects

Comme indiqué dans la section 2.2.1 la simulation d'un effet de salle par convolution avec une réponse impulsionnelle se prête mal aux situations interactives nécessitant la mise à jour des paramètres de l'effet de salle en fonction des contrôles fournis par l'utilisateur ou de sa navigation dans la scène virtuelle. Par ailleurs, même en ayant recours à des algorithmes de convolution opérant sous forme de traitement par blocs dans le domaine spectral, le cot de calcul reste très élevé et, qui plus est, naturellement croissant avec la durée de réverbération de la salle. La charge de calcul peut typiquement atteindre plusieurs dizaines de MIPS (millions d'opérations par seconde de signal) pour une fréquence d'échantillonnage de 48KHz et si l'on désire pouvoir accéder à des durées de réverbération de quelques secondes. De nombreux travaux ont été consacrés au développement d'algorithmes de réverbération artificielle. L'usage de retards rebouclés pour simuler de manière efficace un effet de réverbération est apparu très tôt dans l'histoire des technologies de synthèse audio-numérique [Sch62, Moo79, SP82, Jot97, Ble01]. Cependant, si le principe s'impose de manière intuitive, la compréhension et l'optimisation de ces algorithmes requièrent une étude particulièrement minutieuse sur le plan de la théorie du signal pour simuler de manière fine les caractéristiques de réverbération et éviter la perception d'artefacts (qualifiés généralement de sonorités métalliques). Le lecteur est invité à se reporter aux travaux de J.-M. Jot [Jot97, Jot92b, Jot92a, Jot99, DJ00, JCW97] pour une analyse détaillée du sujet et une description des solutions algorithmiques permettant de respecter les propriétés statistiques du champ réverbéré. Celles-ci sont, en premier lieu, dictées par les observations effectuées sur le comportement physique du champ réverbéré dans un espace clos. En particulier, on peut montrer que, dans une salle, la densité de réflexions D_r (nombre de réflexions par seconde) croît proportionnellement au carré du temps, tandis que la densité modale D_m (nombre de modes par Hz) croît proportionnellement avec le carré de la fréquence:

$$D_r(t) \approx 4\pi c^3 t^2/V, \quad D_m(f) \approx 4\pi V f^2/c, \tag{2.1}$$

où c est la célérité et V le volume de la salle. Ces propriétés, aisément démontrables dans le cas de salles parallélépipédiques sont généralisables au cas de salles de géométrie quelconque [CM82, Kut04]. Ce comportement induit que, dans la partie tardive de la réponse, la séparation entre deux réflexions voisines devient plus petite que le support temporel de chaque réflexion et, à chaque instant, sont ainsi superposées plusieurs réflexions parvenant de différentes directions. De même, dans le domaine fréquentiel, en hautes fréquences, la séparation entre deux modes voisins devient plus faible que la largeur de bande associée à chaque mode et, à chaque fréquence, la réponse de la salle résulte de la superposition de différents modes propres. Ces propriétés se répètent en tout point de la salle avec naturellement des phases et amplitudes différentes pour chaque contribution individuelle. Ainsi le phénomène de réverbération peut être vu comme un processus stochastique où l'on peut s'intéresser tour à tour à la distribution de la réponse fréquentielle ou temporelle en un point donné de l'espace ou au contraire à la distribution spatiale pour une fréquence ou un temps donné. En considérant un seuil minimum de recouvrement fréquentiel entre modes et de recouvrement temporel entre réflexions, les limites de validité de ce modèle statistiques sont définies par la fréquence de Schroeder et le temps de mixage :

$$f_{Schroeder} \approx 2000\sqrt{T_r/V}(Hz), \quad T_{mixage} \approx \sqrt{(V)}(ms) \tag{2.2}$$

La dépendance par rapport au volume V de la salle permet aisément de situer les ordres de grandeur du domaine de validité suivant le type de lieu simulé. Sur le plan algorithmique, la solution adoptée est généralement basée sur des réseaux de retards récursifs ("Feedback Delay Network", FDN) dont la

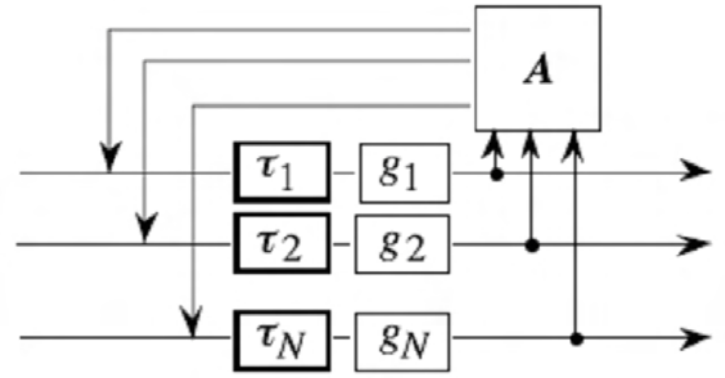


Figure 2.15: Exemple de structure de réseau à retard rebouclé pour simuler un effet de réverbération. La matrice A est une matrice unitaire.

Figure 2.15 présente la structure générale. Le signal à réverbérer alimente en parallèle différentes lignes à retard rebouclées sur elles-mêmes par l'intermédiaire d'une matrice de mélange A. Jot [Jot92a] décrit les conditions de choix des différents paramètres du réseau pour assurer la création d'un bruit gaussien exponentiellement décroissant avec un contrôle sur la dépendance fréquentielle du taux de décroissance.

En premier lieu, la structure de cette matrice est capitale pour assurer le respect des propriétés statistiques mentionnées précédemment. Le choix de la matrice A se porte sur les matrices "unitaires" qui conservent l'énergie (matrices sans pertes) du système et assurent que les modes propres du système sont d'amplitude égale et constante, c'est à dire sans décroissance. Le choix d'une matrice pleine (à coefficients non nuls) permet d'accroître plus rapidement la densité temporelle de réflexions. De même l'homogénéité des coefficients permet d'assurer la convergence vers une distribution gaussienne des amplitudes du ou des signaux engendrés en sortie. L'association de gains g_i permet de gérer la décroissance exponentielle du processus. En réglant le gain g_i de manière conforme au retard auquel il est associé $g_i = \alpha^{m_i}$ (où m_i est égal à la durée du retard exprimée en échantillons), on s'assure d'une décroissance uniforme de l'ensemble des modes. Par extension, on peut également remplacer les gains g_i par des filtres de manière à contrôler la durée de réverbération $T_r(\omega)$ en fonction de la fréquence:

$$20log_{10}|g_i(\omega)| = -60\tau_i/T_r(\omega). \tag{2.3}$$

A partir d'une procédure d'analyse/synthèse décrite dans [JCW97], les filtres gi peuvent être ajustés de sorte à copier très fidèlement les caractéristiques d'enveloppe temps-fréquence, ou relief de décroissance, du signal réverbéré mesuré dans une salle existante en donnant accès au même niveau de fidélité qu'une convolution avec le signal de réverbération original. Dans le cadre d'un modèle paramétrique de salle tel que celui décrit au paragraphe 1.4.3, les filtres gi peuvent être implémentés sous forme de filtres simples autorisant un contrôle en trois bandes. Un tel algorithme de réverbération présente un cot de calcul de l'ordre de 4 MIPS pour un réseau basé sur 8 retards en parallèle. Outre le gain par rapport à une convolution, l'avantage réside dans la facilité de contrôle de la décroissance et l'indépendance du cot de calcul en fonction de la durée de réverbération.

Structure algorithmique associée au modèle générique de salle

La Figure 2.16 montre la structure algorithmique d'un processeur de spatialisation associé au modèle générique d'effet de salle présenté au paragraphe 1.4.3. Conformément au découpage en sections temporelles du modèle, l'algorithme est constitué de la mise en cascade de différents modules réalisant

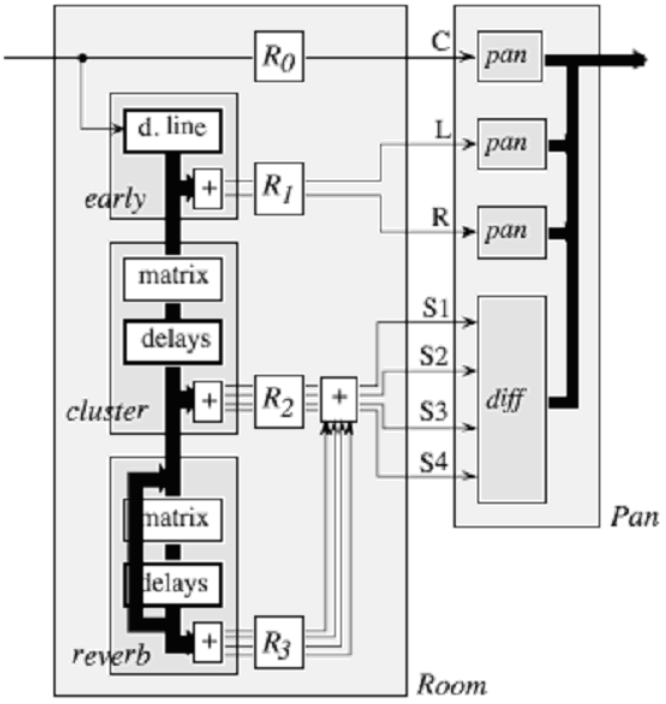


Figure 2.16: Structure algorithmique du processeur associé au modèle d'effet de salle Générique.

la synthèse des premières réflexions et des réflexions tardives qui alimentent à leur tour un module réverbération tel que celui décrit précédemment. Chacun des modules est basé sur une ou plusieurs lignes à retard associées à des gains ou filtres autorisant le réglage du niveau et de la dépendance fréquentielle des différentes sections temporelles de l'effet de salle. Cette structure permet de façonner la distribution temps-fréquence de l'effet de salle et produit en sortie un ensemble de signaux véhiculant respectivement le son direct et différents canaux pour les premières réflexions et les signaux de réverbération. La distribution spatiale de l'effet de salle est synthétisée, dans un second étage du traitement, par l'encodage des effets directionnels associés à chacune des sections temporelles et selon le format de restitution sélectionné. Selon les exigences de résolution spatiale mentionnées dans le paragraphe 1.4.3, l'encodage spatial doit, autant que possible, répondre à une précision maximale pour le son direct, tandis que cette contrainte peut être relâchée pour les sections plus tardives. Le traitement spatial des signaux de réflexions tardives et de réverbération vise à respecter les propriétés de décorrélation interaurale associées au champ diffus.

Par l'ajustement des degrés de liberté du modèle générique, l'algorithme permet de modifier les attributs perceptifs utilisés pour décrire la qualité acoustique de l'effet de salle produit. Inversement on peut concevoir une interface de commande de haut niveau proposant de contrôler l'effet de salle produit selon ces mêmes attributs perceptifs. Cette interface suppose de pouvoir inverser les expressions reliant les attributs perceptifs aux paramètres de la distribution temps-fréquence-espace de l'effet de salle. De paramètres descripteurs de la qualité acoustique d'une salle, ces attributs deviennent alors prescripteurs de l'effet de salle conçu par l'auteur de la scène virtuelle.

2.3 Structured audio rendering and perceptual optimisations

Comme nous l'avons vu dans le paragraphe précédent, le rendu d'une source sonore 3D implique un nombre important d'opérations de traitement du signal. Même dans le cas de modèles simplifiés, effectuer l'ensemble de ces traitements pour un grand nombre de sources sonores reste trop coteux en temps de calcul pour la plupart des applications. De plus, les solutions de rendu hardware [EAX04] (voir Chapitre 3) ne supportent qu'un nombre limité de sources sonores simultanées, aussi appelées "voies" (jusqu'à 128 pour génération actuelle). Il est pourtant clair qu'un grand nombre de sources sonores peut être nécessaire au rendu d'un environnement réaliste. D'une part, on veut pouvoir représenter des sources sonores étendues (un train par exemple) ou complexes ce qui nécessite l'utilisation de plusieurs sources. D'autre part, le rendu de chemins de propagation précoces, nécessite également le rendu de nombreuses sources secondaires. Enfin, dans certaines applications, comme le jeu vidéo, une musique d'ambiance peut-être également rendue sous forme spatialisée en utilisant un ensemble de sources sonores 3D spécifique. Une problématique qui apparaît très vite est alors de pouvoir rendre efficacement un grand nombre de sources que ce soit de manière logicielle ou en effectuant dynamiquement un mapping sur un nombre de canaux matériels limité ("voice management"). Dans cette section, nous présentons différentes stratégies permettant de structurer une scène sonore et la représenter à différent niveaux de résolution afin de pouvoir maîtriser dynamiquement la complexité du rendu. La structuration d'une scène sonore et la perception de sources sonores multiples ont fait l'objet de nombreuses recherches tant dans le domaine de la psycho-acoustique que de la communauté de l'analyse de scène sonores [Bre90, BvSJC05, BSK05]. Comme nous le verrons, une particularité de ces approches est de s'adapter au contenu des signaux devant être spatialisés ainsi qu'aux propriétés de l'auditeur humain. En pratique, maîtriser la complexité du rendu audio 3D implique de traiter principalement trois aspects : gérer l'importance relative des différentes sources sonores dans la scène, gérer la complexité spatiale de la scène, gérer la complexité dans le traitement du signal.

2.3.1 Perceptual importance of sound sources and auditory masking

La notion d'importance de source sonore est fondamentale pour la structuration et l'optimisation des traitements. Elle peut intervenir pour guider différents types de simplifications de la scène sonore. En outre, le tri par importance des sources sonores est la technique la plus utilisée pour rendre un grand nombre de sources avec un nombre de voix matérielles limitées. Pour n voix, on rend simplement les n sources les plus importantes à chaque trame audio. Une question fondamentale est alors de définir une bonne métrique d'importance. Les métriques les plus couramment utilisées estiment l'atténuation des différentes sources sonores dans la scène (e.g., due à la distance, aux occultation, etc.), éventuellement combinées avec une notion de durée des sons (un son dont la majeure partie a déjà été jouée peut être interrompu plus facilement). Enfin, l'utilisateur peut être libre de moduler cette valeur pour donner plus d'importance à certains sons. Il est clair que dans l'hypothèse où les sons sont à peu près similaires en terme de niveau ou sonie, cette approche peut donner des résultats satisfaisant de manière très efficace. Néanmoins, dans la plupart des cas, elle peut conduire à une solution perceptivement sub-optimale dont la qualité va se dégrader fortement lorsque le nombre n de sources jouables simultanément diminue. Afin de limiter ces problèmes, on peut s'inspirer de deux constatations. Tout d'abord, les variations d'énergie acoustique au cours du temps dans un même signal peuvent être très importantes. En général, l'énergie varie de manière rapide et discontinue en comparaison de l'atténuation liée à des critères géométriques qui elle, varie continuement et lentement lorsque la source se déplace. En conséquence, ces variations peuvent être largement plus importantes que les atténuations des sources dont la plupart se trouvent, en général, dans un périmètre limité autour de l'auditeur, et sont donc atténuées de manière similaire.

L'énergie instantanée du signal est alors un bien meilleur descripteur de son importance. La combinaison de l'énergie instantanée du signal émis avec l'atténuation liée à la propagation est donc un bon critère pour définir l'importance d'une source sonore. Des travaux récents sur la synthèse progressive de mixtures sonores utilisant ce principe confirment cette hypothèse [GLT05, Tsi05a]. L'énergie instantanée ainsi que d'autres indicateurs des propriétés du signal peuvent en outre être pré-calculés. De manière similaire à des standards comme *MPEG7* et les travaux en indexation de bases de données audio [HSP99, Log00, Pee04], ces descripteurs peuvent être stockés dans une représentation étendue des signaux sonores avec un impact très limité sur la mémoire requise [TGD04]. Au final, cette méthode reste donc très efficace tout en s'adaptant aux caractéristiques des signaux à traiter.

D'autre part, lorsque plusieurs sources sonores simultanées nous parviennent, il est très peu probable que nous les percevions toutes. En effet, des phénomènes de masquages auditifs complexes entrent alors en jeu. Comme cela a été le cas en compression du son (avec des standards comme mp3 par exemple), diverses approches ont été développées pour mettre à profit ces phénomènes afin d'optimiser le rendu sonore ou la synthèse sonore, en supprimant les parties de la scène sonore qui ne seront pas entendues. Là encore, on peut faire le parallèle avec les approches d'élimination des parties cachées utilisées pour optimiser le rendu graphique 3D interactif. Lagrange et Van Den Doel [vdDPA$^+$02, LM01, vdDKP04] par exemple, ont proposé d'utiliser un modèle de masquage acoustique pour accélérer un algorithme de synthèse modale en supprimant les modes inaudibles. De manière similaire dans [TGD04] des algorithmes ont été proposés pour estimer de manière efficace les sources sonores audibles (resp. masquées) d'une scène sonore. Cet algorithme glouton commence par trier les sources par importance (dans [TGD04] un indicateur de sonie est utilisé). Puis les sources sont considérées par ordre d'importance décroissante jusqu'à ce que leur somme masque la somme des sources restantes. La tonalité du signal, un autre indicateur déterminant si le signal est proche d'un bruit ou proche d'un signal harmonique, peut être également utilisé pour ajuster plus finement les seuils de masquage acoustique [Ran01, KAG$^+$02]. L'algorithme détermine donc dynamiquement le nombre de sources audibles à chaque trame de rendu pour ne rendre que celles-ci, et ce de manière perceptivement transparente pour l'auditeur. Il a été également appliqué avec succès à l'optimisation de calculs de réverbération par convolution avec de longues réponses impulsionnelles en découpant le filtre en petits blocs et en considérant chaque bloc comme une source sonore distincte devant être mixée [GLT05, Tsi05a]. Bien entendu, la mesure de l'importance d'une source sonore ne se limite pas forcément à des critères purement énergétiques. Des critères de type contraste ou urgence [ELD91, HC95] ou bien d'autres critères cognitifs de plus haut niveau [Bre90] pourraient être également utilisés pour quantifier l'importance relative des différentes sources sonores de l'environnement pour en adapter le traitement.

2.3.2 Spatial level of detail and sound source clustering. Auditory *impostors*

La gestion de la complexité spatiale de la scène est un aspect très important pour le rendu audio 3D. En effet, un grand nombre d'effets et en particulier les traitements de spatialisation dépendent de la position des différentes sources sonores dans l'espace. Or, notre perception spatiale du son a des limites (e.g. masquages frequentiels et temporels, précision de la localisation sonore, etc.) [Moo97, Bla97, BvSJC05, BSK05] que l'on peut mettre à profit afin de créer des représentations simplifiées de la scène sonore. Ceci est d'autant plus justifié que le nombre d'évènements sonores simultanés est grand, puisque l'on peut ne consacrer qu'une attention limitée à chacun, voire à un sous ensemble seulement [BvSJC05]. A cette fin, plusieurs approches ont été développées afin de créer des représentations hiérarchiques d'une scène sonore. A ce titre, elles peuvent être mises en parallèle avec les algorithmes de niveau de détail, utilisés en graphique pour simplifier la géométrie 3D à afficher [LRC$^+$02].

Une première famille d'approche opère par regroupement des sources sonore proches pour former

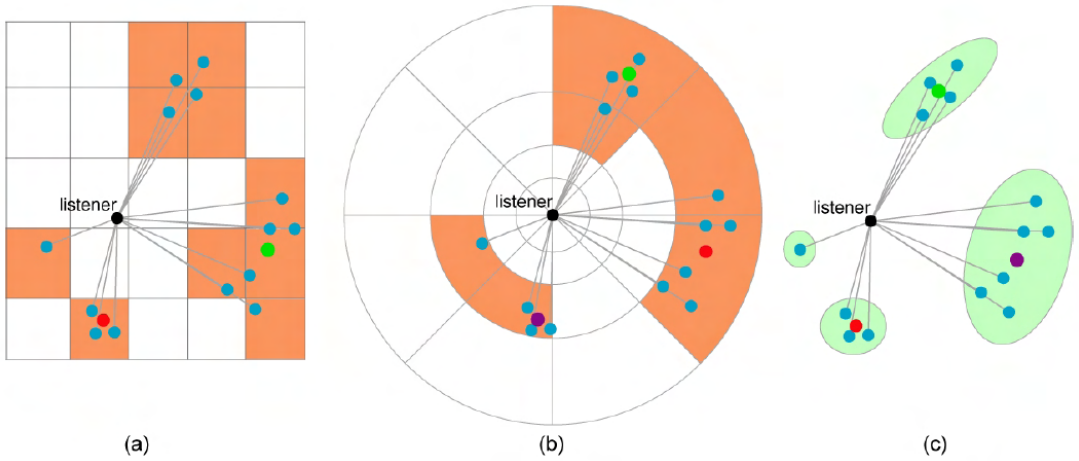


Figure 2.17: Trois exemples de regroupement de sources sonores. (a) Utilisation d'une structure régulière. (b) d'une structure à dé-raffinement progressif et (c) d'une technique de regroupement adaptatif.

des imposteurs sonores. Un imposteur sonore permet alors de représenter plus simplement le groupe de source considéré. Par souci de compatibilité avec des approches de rendu standard, ces imposteurs peuvent être construits comme un sous-ensemble de sources ponctuelles représentatives de la scène sonore initiale. Chaque groupe de sources est alors remplacé par une unique source représentative dont la position, en général le barycentre du groupe, peut être adaptée au cours du temps en fonction de l'importance des diverses sources dans le groupe [TGD04]. Il est également nécessaire de déterminer un signal équivalent pour l'imposteur sonore, par exemple la somme des signaux de chaque source dans le groupe. Le regroupement des sources peut être mis en uvre de diverses manières en particulier en utilisant une subdivision spatiale ou directionnelle fixée [Her99, MW04] ou bien par des algorithmes de regroupement adaptatifs, de type *k-means* [TGD04] (Figure 2.17). Les algorithmes de regroupement adaptatif présentant plusieurs avantages : ils peuvent produire un nombre de groupes cibles, ils concentrent leur résolution là où elle est nécessaire et peuvent être contrôlés par une variété de métriques d'erreur. En particulier, l'importance des signaux sonores peut, là encore, être utilisée pour contrôler le regroupement des sources [TGD04]. On peut noter que cette approche s'apparente à certaines techniques de compression de contenu audio spatialisé (e.g., enregistrements "surround" pour le cinéma). Un exemple d'une telle technique est le "binaural cue coding" (BCC) [BF03, FB03, FM04] qui extrait des indices de localisation spatiale à partir d'un enregistrement multi-canal et encode le résultat comme une mixture mono et un fiux de positions dans l'espace qui évolue dans le temps. A l'arrivée chaque trame est décodée et re-spatialisée en fonction de la position déterminée par l'encodage. On peut mettre en parallèle une telle stratégie avec la définition d'un groupe de sources unique dont la position évolue avec le temps, d'une manière similaire à [TGD04]. Bien évidemment, dans le cas du BCC qui résoud un problème inverse en partant de la mixture finale, on ne dispose pas directement de la position des sources sonores individuelles comme c'est le cas dans un système de spatialisation traditionnel. Comme nous le verrons dans la section 1.6, associer une position 3D à un enregistrement est une problématique qui peut intervenir également pour le rendu audio 3D direct à partir d'enregistrements. Là encore, des critères spatiaux seuls peuvent ne pas suffire à segmenter la scène sonore de la manière la plus perceptivement efficace. L'analyse de scène sonore [Bre90] propose d'autres critères de regroupement d'évènements sonores (simultanéité, principe proches de la théorie de la Gestalt) . D'autres approches, exploitent des

représentations mathématiques qui encodent les propriétés directionnelles du champ sonore, par exemple par décomposition sur une base d'harmoniques sphériques. Mises en uvre dans la technique d'encodage et restitution Ambisonics [MM95] (voir également Chapitre 3), ces approches permettent un niveau de détail par troncation de la décomposition en harmoniques, ce qui entraine une diminution de la précision spatiale du rendu (i.e., un filtre passe-bas dans l'espace des directions). Elles permettent également des opérations globales de rotation par exemple sur un groupe de sources encodées dans cette représentation. Nous y reviendrons dans la dernière section de ce chapitre. Ce type de représentation peut être utilisé pour représenter des imposteurs sonores non ponctuels avec une résolution spatiale variable ou pour recréer l'arrière plan sonore d'une scène, les sources de premier plan étant, elles, représentées comme des sources ponctuelles. Une telle approche s'apparenterait alors aux techniques de type, "environnement map" ou "cube-map" en graphique [FVFH90].

2.3.3 Progressive signal representations and processing scalability

Une autre possibilité pour permettre un rendu à différents niveaux de détail d'une scène sonore est de travailler directement avec une représentation multi-échelle ou progressive des signaux sonores sources. Dans ce cas, on peut imaginer effectuer l'ensemble des traitements requis pour toutes les sources. Si l'on dispose d'une représentation multi-échelle des signaux, on pourra garantir un budget d'opérations fixé, par exemple de manière à ce que chaque source ne contribue au résultat final qu'au prorata de son importance. Une possibilité est ainsi d'encoder le signal en ondelettes [DDS02], ou bien d'utiliser une représentation fréquentielle en espace de Fourier [Tsi05b]. Une autre famille d'approches propose également d'effectuer le traitement directement sur des signaux compressés à l'aide d'un codec perceptif (type mp3), ce qui peut être plus efficace qu'un cycle décodage, traitement, ré-encodage. Néanmoins, un décodage partiel doit en général être effectué et les traitements en domaine codé sont en général plus délicats et nécessitent des filtres adaptés [Tou00, TEP04]. Comme on peut le constater, la séparation entre compression et traitement du signal audio tend donc à s'estomper pour aboutir à des approches dans lesquelles la représentation des signaux est adaptée à la fois à la transmission et au traitement. Cette problématique est particulièrement importante pour des applications de rendu audio distribué dans le cadre d'application massivement multiutilisateurs par exemple.

2.4 Rendering from spatial recordings

Les précédentes sections de ce chapitre se sont concentrées sur des approches de rendu audio classiques où l'environnement sonore est décrit comme un ensemble de sources ponctuelles. Une autre possibilité, qui se prête bien à la capture et la restitution d'environnements complexes et réaliste est de réaliser un rendu sonore directement à partir d'enregistrements spatiaux. Contrairement au cas classique dans lequel chaque source émet un signal monophonique, les signaux d'entrée sont en général multi-canaux et correspondent à un échantillonnage spatial de l'environnement sonore. Comme les techniques de rendu à base d'image en graphique qui proposent d'échantillonner la fonction plénoptique [AC01, GGSC96, LH96], ces techniques échantillonent une fonction "plénacoustique" correspondant au champ sonore. A partir de cet échantillonnage plusieurs applications sont possibles comme la restitution spatiale, l'interpolation du point d'écoute ou le déplacement des sources sonores. En pratique toutefois, le grand nombre de canaux requis en limite souvent l'applicabilité.

2.4.1 Coincident recordings and directional decompositions

Une première catégorie d'approches utilise une série d'enregistrements concidents permettant d'obtenir une décomposition directionnelle du champ sonore par décomposition sur une base de fonctions directionnelles (harmoniques sphériques par exemple). Ce type de décomposition est à la base du format d'enregistrement Ambisonics que nous avons déjà évoqué précédemment (voir egalement autre article). L'intérêt d'un tel format est qu'il permet bien entendu la restitution spatiale mais rend possible l'interpolation entre enregistrements. De plus, il existe des solutions commerciales permettant de réaliser des enregistrements au format-B [Sou] (une décomposition à l'ordre 1 sur une base d'harmoniques sphériques). Bien qu'ayant une résolution spatiale limitée, ce type d'enregistrement peut être utilisé sur le terrain et permet d'obtenir des backgrounds sonores très riches. En outre, il est possible de manipuler par simple combinaison linéaire les enregistrements pour effectuer des rotations du champ sonore. D'autres effets, comme des changements de perspective sont également possibles [Dan00, Lar01] ainsi que des effets de transition entre de tels enregistrements réalisés en différents points. Avec le développement de microphones d'ordre supérieur [ME04a, TRI], cette approche pourra permettre une meilleure résolution spatiale, au prix d'un plus grand nombre de canaux à traiter.

2.4.2 Non-concident recordings

Dès lors que l'on désire obtenir une représentation spatiale de l'acoustique d'un lieu, il est possible d'échantillonner la fonction plénacoustique avec un nombre d'enregistrements omnidirectionnels non-concidents. Ainsi, d'une manière similaire à l'interpolation de vue en graphique [CW93a, BBM$^+$01, HAA97], Radke and Rickard [RR02] ont proposé une approche visant à interpoler de manière physiquement réaliste le signal audio le long d'une ligne reliant deux microphones. A partir d'une représentation temps-fréquence des deux enregistrements, ils utilisent une technique dérivée d'une approche de séparation aveugle de source sonore [JRY00] pour associer une position à chaque composante fréquentielle au cours du temps (sous hypothèse d'une mixture anechoque des signaux sources et que les transformées de Fourier à court terme des sources ne se recouvrent pas). Cette position est ensuite utilisée pour resynthétiser un signal à une position virtuelle le long de la ligne reliant les microphones en supposant que les sources sont simplement retardées par le délai de propagation et atténuées par l'inverse de la distance. D'autres approches [AV02, Do04] proposent d'échantillonner spatialement de manière massive la fonction plenacoustique et de l'interpoler directement. A partir de l'étude de la transformée de Fourier de la fonction plénacoustique, ces études ont proposées des bornes sur le taux d'échantillonnage spatial pour une reconstruction sans aliassage. Pour des fréquences de l'ordre de 5kHz, l'écartement entre deux prises de son doit être de l'ordre de la dizaine de centimètres, pour 10 kHz de l'ordre du centimètre. Ceci limite fortement l'usage pratique de ces méthodes.

2.4.3 Extracting structure from recordings

Une dernière possibilité pour le rendu à partir d'enregistrements consiste à effectuer une analyse de la scène sonore afin d'abstraire des données de plus haut niveau (signaux et positions des différentes sources sonores par exemple). Ces données peuvent alors permettre une modification plus fine de la scène lors de la restitution (déplacement ou altération des sources, déplacements de l'auditeur, etc.) De nombreuses approches peuvent être utilisées à cette fin, qui proviennent de champs d'application différents. Une première solution est d'appliquer des techniques de séparation aveugle de signaux [JRY00, BZ03, VRR$^+$03, Ave03], comme nous l'avons évoqué précédemment. Ces techniques sont couramment utilisées pour la séparation de locuteurs en télécommunication, pour traiter des enregistrements musicaux (e.g.,

karaoké) ou pour “up-mixer” de manière aveugle des signaux stéréophoniques en format 5.1 par exemple. D’autres technique que nous avons déjà évoquées s’intéressent également à extraire des indices de localisation binauraux à partir d’une bande son stéréophonique ou multi-canal [BF03, FB03, FM04]. On peut alors séparer, dans une certaine limite, les différentes sources de la scène afin de les modifier, supprimer, déplacer, etc. Cette séparation peut être explicite ou non. Si le but est de modifier légèrement la scène, il est possible d’identifier et modifier les composants fréquentiels de chacune des sources et de resynthétiser directement une mixture modifiée [Ave03]. Là encore, ce type d’extraction de structure rejoint la problématique de l’analyse de scène sonore (“Auditory Scene Analysis”) [Bre90] qui vise à reproduire un modèle complet de notre cognition auditive, de manière similaire à la vision par ordinateur pour l’image.

Chapter 3

Interfaces for spatial audio reproduction

Dans ce chapitre nous nous intéressons aux techniques logicielles et aux dispositifs matériels associés permettant de réaliser une restitution spatialisée du son pour les environnements virtuels. La spatialisation du son, c'est-à-dire la simulation des indices de localisation spatiale de notre audition pour des sources sonores virtuelles, est l'un des thèmes de recherche les plus actifs en acoustique à l'heure actuelle. Il faut dire que les applications grand-public sont légions : jeux vidéo, home cinéma, musique, que ce soit à la maison ou sur des dispositifs nomades. Dans les applications de réalité virtuelles, l'ajout de son spatialisé contribue également à renforcer la sensation d'immersion [LVK02]. Ces technologies reposent toutes sur d'intenses recherches fondamentales sur la perception spatiale du son chez l'homme. Des ouvrages et travaux de références comme [Beg94, Bla97] peuvent également être consultés pour plus de détails. Les systèmes de restitution sonore spatiale vont, d'une manière générale, s'intéresser à reproduire aux oreilles d'un auditeur les indices de localisation appropriés pour une source sonore virtuelle en fonction de sa position souhaitée. On peut catégoriser ces systèmes en deux familles. La première série d'approches, que l'on peut qualifier de perceptives, repose sur le fait qu'un modèle simple de différence de niveau sonore ou de temps d'arrivée aux deux oreilles suffit à créer l'illusion de positionnement d'une source "fantôme" entre les positions physiques des haut-parleurs. Ce principe est en particulier à la base de la restitution stéréophonique classique mais peut être étendu à des ensembles de haut-parleurs plus complexes. La deuxième série d'approches vise à reconstruire de manière physiquement précise le champ acoustique au voisinage de l'auditeur. Dans cette catégorie, se distinguent les approches binaurales qui reconstituent le champ sonore en deux points (les oreilles de l'auditeur) et des approches plus globales, reproduisant le champ sonore correct dans toute une zone de l'espace. Chacune de ces approches a ses avantages et ses inconvénients en terme de ressources de calcul, complexité du dispositif et nombre d'auditeurs simultanés. Pour un complément d'information, on peut également se référer aux ouvrages suivants [Beg94, PB04a]. Nous présentons également dans ce chapitre quelques exemples d'intégration de dispositifs de restitution spatiale du son dans des systèmes de réalité virtuelle immersifs. En particulier, nous dressons un comparatif des différentes approches et de leurs conditions d'utilisation optimale. Enfin, nous consacrons la dernière section de ce chapitre à un bref tour d'horizon des quelques librairies logicielles existantes permettant d'implémenter de telles techniques et bénéficiant pour certaines d'une accélération matérielle.

3.1 Perceptual approaches and phantom sources

La première famille d'approches pour la restitution spatiale du son s'appuie sur un contrôle simple des indices de localisation inter-auraux à partir d'un ensemble de deux haut-parleurs ou plus, situés autour

de l'auditeur dans l'espace de restitution.

3.1.1 Stereophony

Le modèle de restitution spatiale du son le plus utilisé encore aujourd'hui reste la stéréophonie, utilisant un couple de haut-parleurs situés devant l'auditeur [ste89, SE98]. En contrôlant le retard ou le gain relatif des signaux envoyés à chaque haut parleur, il est possible de recréer une version simplifiée des indices de localisation inter-auraux, la différence de temps d'arrivée (ITD) et la différence de niveau (ILD) aux deux oreilles. Cette restitution des indices de localisation permet de créer une source sonore fictive dont la direction de provenance peut être contrôlée librement dans la zone encadrée par les deux haut-parleurs. Il est ainsi possible de contrôler soit le retard relatif des deux canaux, soit l'intensité relative, soit les deux simultanément. En général, c'est cette dernière solution qui est préférée et on la retrouve dans les dispositifs de prise de son les plus courants comme les paires stéréophoniques "AB" de l'ORTF par exemple, composée de deux microphones directionnels (pour la différence d'intensité) et non concidents (pour la différence de temps d'arrivée). D'autres alternatives, comme le "DECCA Tree", utilisent trois microphones omnidirectionnels [SE98, Stra] et se basent donc principalement sur les différences de temps d'arrivée. L'utilisation d'une unique différence de délai entre les deux canaux peut conduire à une zone de restitution très limitée. En effet, notre audition spatiale est très sensible au temps d'arrivée des fronts d'ondes, donnant la priorité aux fronts d'ondes qui nous atteignent en premier. Cette loi psycho-acoustique est aussi appelée effet Hass ou loi du premier front d'onde [Moo97]. Une conséquence est que si l'auditeur se décale, même légèrement, de l'axe médian des deux haut-parleurs, l'ensemble de la scène stéréophonique tend à se "replier" de manière assez abrupte sur la position du haut-parleur le plus proche. L'utilisation d'une différence de gain contribue à limiter cet effet. Dans le cas du DECCA Tree, un troisième microphone en position centrale et avancé vers la source sonore met à profit ces effets pour stabiliser, lors de la restitution, l'image centrale entre les haut-parleurs. De plus, un mixage monophonique de deux signaux uniquement décalés dans le temps conduit à des interférences particulièrement audibles, ce qui a historiquement limité l'utilisation de ces techniques dans la production audio-visuelle. Là encore, l'introduction d'un gain inter-canal en plus du retard, limite ces problèmes. La plupart des mixages stéréophoniques réalisés en studio n'utilisent de fait qu'une simple différence d'intensité entre les canaux, que l'on retrouve dans ce milieu sous le nom de "pan-pot" (de l'anglais, "panoramic potentiometer"). Des dispositifs à base de couples de microphones concidents (M/S, X/Y) [SE98] permettent d'effectuer une prise de son sur le terrain qui respecte ce modèle de mixage. Bien que ne permettant de restituer une source que dans le plan horizontal et devant l'auditeur, la stéréophonie reste une technique très efficace pour la restitution sonore. Pour une application où la localisation tri-dimensionnelle complète (i.e., haut/bas, avant/arrière) n'est pas nécessaire, elle demeure une excellente solution. Des études ont en effet montré qu'une tche de localisation auditive dans un plan horizontal, dans laquelle l'auditeur peut se déplacer virtuellement dans l'environnement, n'est que très peu améliorée par l'utilisation d'une approche binaurale plus complexe [LGST00]. On peut supposer alors que l'utilisateur met à profit les informations relatives liées à ses déplacements (variation d'intensité ou de direction des sources, effet Doppler) pour améliorer sa navigation et en particulier déterminer l'instant où une source vient d'être dépassée.

3.1.2 Multi-channel systems

Par extension des techniques stéréophoniques classiques, il est possible de reproduire un son spatialisé sur des ensembles de haut-parleurs, plans ou tri-dimensionnels, situés autour de l'auditeur. L'exemple le plus utilisé actuellement de ce type de dispositifs est la configuration "5.1" ou "7.1" des formats de

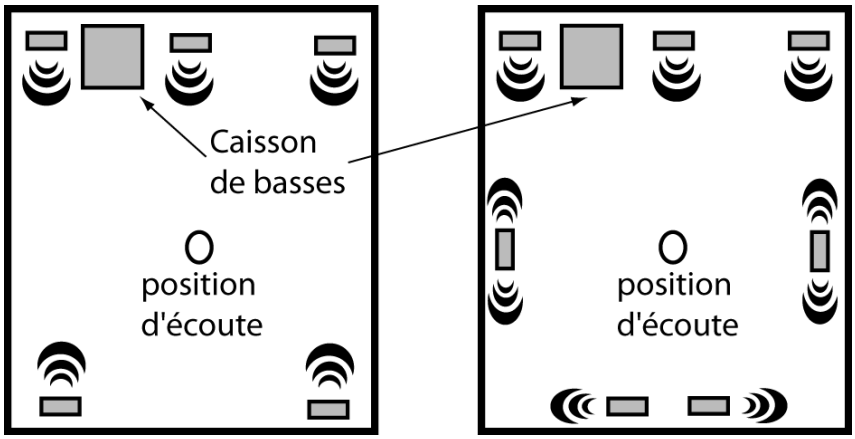


Figure 3.1: Systèmes de reproduction de type cinéma. A gauche, un système classique ”5.1”. A droite, un système ”7.1”.

restitution sonore cinématographiques (Figure reffig:surround). Ces systèmes comprennent classiquement trois canaux à l'avant et deux à l'arrière, le canal central avant étant principalement destiné à ancrer l'image acoustique des dialogues au centre de l'écran. Ce canal de restitution compense l'effet de précédence pour les auditeurs situés sur les côtés et évite une perte d'intelligibilité liée au repliement des dialogues sur le canal droit ou gauche. Un système 7.1 modifie cette configuration par le décalage des enceintes arrière en position latérale et l'ajout d'un canal arrière central. Dans ce cas, des enceintes latérales dipolaires (irradiant à la fois à l'avant et à l'arrière en opposition de phase) sont préférées pour une meilleure intégration des effets avant et arrière. L'utilisation de deux haut-parleurs proches (“stéréo dipole”) pour le canal arrière central évite des confusions de localisation avant/arrière dans le plan médian. Les deux systèmes disposent en outre d'un canal de restitution dédié aux basses fréquences (typiquement 80 Hz et moins) qui sont reproduits par un ou deux caissons de grave, les basses fréquences étant moins facilement localisables.

Par extension, ce type de configuration est largement utilisé dans toutes les applications 3D interactives grand-public comme les jeux vidéo par exemple. Différentes stratégies sont alors possibles pour déterminer les gains (et éventuellement retards) relatifs à appliquer aux différents canaux pour une restitution perceptivement optimale [DFMM99]. Parmi les approches les plus courantes, on peut également citer VBAP (”Vector-Based Amplitude Panning”) [Pul97] qui utilise un système de haut-parleurs répartis dans l'espace autour de l'auditeur. Le triplet de haut-parleurs encadrant la direction à reproduire est alors utilisé, les gains de chaque haut-parleur étant simplement déterminés par les produits scalaires de la direction des haut-parleurs avec la direction souhaitée (Figure 3.2). Malgré l'existence d'un certain nombre de dispositifs ”empiriques” (par exemple, ”Surround DECCA Tree” [Strb]), l'extension des techniques de prise de son à un système multi-canal est difficile. Seules des techniques basées sur une analyse mathématique plus complexe du champ sonore, peuvent conduire à un dispositif réellement approprié, comme les systèmes Soundfield [Sou] ou Trinnov Surround Recording Platform [TRI] sur lesquels nous reviendrons.

3.2 Reconstructing a physical sound field

La deuxième famille d'approches pour la restitution spatiale du son repose sur des fondements physiques plutôt que perceptifs et vise à reconstruire aux oreilles de l'auditeur le champ sonore qui serait créé par

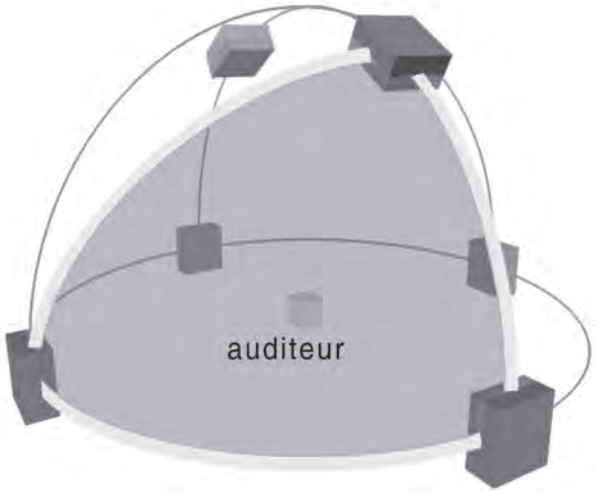


Figure 3.2: Les techniques de type VBAP généralisent le “pan-pot” d’intensité à des configurations arbitraires en considérant des triplets de haut-parleurs.

les sources sonores dans l’espace de restitution si celles-ci étaient réelles. Contrairement aux approches précédentes, elles permettent également de définir des systèmes de prise de son duaux, adaptés au mode de restitution. Ce dernier point est particulièrement utile pour la restitution de scènes sonores 3D réalistes enregistrées sur le terrain, un peu à la manière des techniques de ”rendu à base d’images” en graphique.

3.2.1 Binaural techniques

Les techniques de restitution dites binaurales visent à reproduire le champ sonore directement aux oreilles de l’auditeur à partir d’un couple de signaux. Le dispositif de restitution privilégié pour les techniques binaurales est donc le casque stéréophonique [Møl92]. La connaissance des interactions de l’onde sonore avec la partie supérieure de notre corps [KG83, WK89] permet de définir une paire de filtres qui peuvent être appliqués à un son monophonique en fonction de la direction d’incidence souhaitée lors d’une restitution stéréophonique. Ces filtres spécifiques sont couramment référencés dans la littérature sous le terme “fonction de transfert de la tête” (Head Related Transfer Functions ou *HRTFs* en anglais). Déterminer et échantillonner spatialement les HRTFs est donc un point clé des techniques de spatialisation binaurales. En outre, les HRTFs sont dépendantes de la morphologie de l’auditeur, qui a une infiuence importante sur certains indices spectraux nécessaires à une localisation correcte. S’il est possible, et d’ailleurs courant, d’utiliser des HRTFs “génériques”, celles-ci peuvent entraner des erreurs de localisation significatives, en particulier des confusions avant/arrière. Dans le cadre d’applications de réalité virtuelle, l’ajout d’un système de suivi de mouvement de la tête (par exemple à l’aide d’un capteur d’orientation sur le casque) améliore la localisation et aide également à réduire les confusions avant-arrière [BWA01].

Modeling the “Head Effect” and adaptation to the listener

Historiquement, des modèles simplifiés de la tête (par exemple, une sphère) ont été utilisés pour obtenir une modélisation analytique des HRTFs [AAD01, DAA99]. Ces modèles ont été étendus pour prendre en compte des effets plus subtils comme l’infiuence du buste (réflexions sur les épaules par exemple) [AAD99, BD98]. Toutefois, à l’heure actuelle la plupart des HRTFs utilisées dans les systèmes de spatialisation sont directement obtenues à partir de mesures en chambre anéchoique, que ce soit sur

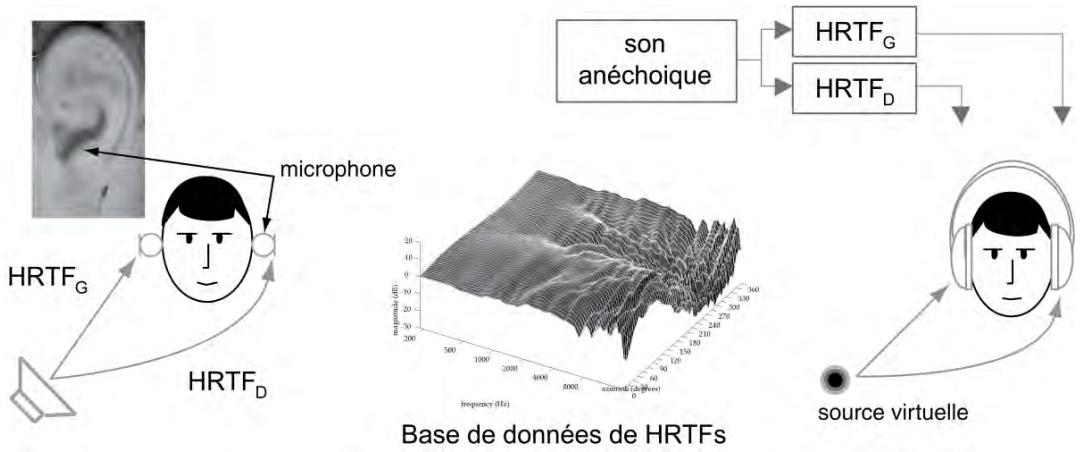


Figure 3.3: Principe de la restitution sonore binaurale. Les indices de localisation sonore spatiale sont caractérisés par des paires de filtres appelés HRTFs, qui peuvent être mesurées directement aux oreilles d'un auditeur en déplaçant une source dans l'espace autour de lui (à gauche). Ces mesures permettent de réaliser une base de données de HRTFs pour différentes directions d'incidence (au centre). La spatialisation d'un son monophonique est alors effectuée par filtrage du signal par la paire de filtres correspondant à la direction d'incidence souhaitée (à droite).

des têtes artificielles ou directement sur des sujets, en plaçant des petits microphones à l'entrée ou à l'intérieur du canal auditif (Figure 3.3). De ces mesures, des modèles permettant une implémentation efficace du traitement peuvent être dérivés comme nous le décrirons par la suite. La mesure acoustique des HRTFs est une opération peu aisée qui nécessite des équipements très spécifiques et qui est sujette à de nombreux problèmes potentiels (bruit de mesure, répétabilité, etc.) [MSHJ95, MHJS95, WK05]. Plusieurs bases de données de mesures de HRTFs sont néanmoins disponibles en ligne, en particulier KEMAR [GM94, kem], CIPIC [cip], AUDIS [aud], LISTEN [lis] et ont été utilisées pour de nombreuses applications. Le problème fondamental de l'utilisation de HRTFs pour la spatialisation réside dans le fait que ces filtres sont fortement dépendants de la morphologie de l'auditeur [TNKH98]. Plusieurs approches ont donc été proposées pour adapter un jeu de filtres génériques à un auditeur particulier. Par exemple, à partir de mesures physiques des paramètres morphologiques (taille de la tête, taille du pavillon de l'oreille, etc.), il est possible de légèrement déformer la réponse spectrale des filtres de manière à aligner certains pics ou creux sur les fréquences appropriées [Lar01]. Une autre approche courante consiste à construire le meilleur jeu de HRTFs pour un auditeur spécifique à partir d'une base de données. On peut ainsi demander à l'auditeur de procéder à une "calibration" préalable en choisissant les paires de HRTFs donnant le meilleur rendu d'un ensemble donné de positions de sources virtuelles [SF03]. Une dernière possibilité qui permet plus de fiexibilité dans la génération des filtres est la simulation numérique [KNPC99, Kat01a, Kat01b]. A partir d'un maillage tri-dimensionnel de la tête et du torse, des méthodes d'éléments finis permettent d'obtenir les fonctions de transfert souhaitées (voir Figure 3.4). La difficulté dans ce cas reste les temps de calculs importants et l'acquisition d'un maillage du buste de l'utilisateur. En effet, celle-ci nécessite l'utilisation d'un scanner 3D et un traitement important pour pré-conditionner les maillages obtenus. Au final, l'acquisition de modèles 3D reste donc actuellement tout aussi contraignante que la mesure acoustique mais pourrait permettre dans l'avenir de simuler des HRTFs individualisées de manière beaucoup plus fiexible et dans des contextes variés. Un autre problème de la restitution binaurale provient du dispositif de restitution lui-même. En effet, lorsqu'un utilisateur porte un casque stéréophonique, une cavité acoustique se forme entre le haut-parleur et le tympan. Les

résonances de cette cavité et la réponse du casque lui-même induisent un filtrage supplémentaire qui peut infiuencer la restitution d'une manière significative [MHJS95]. Une conséquence est la difficulté dans ce cas à externaliser le son qui est très souvent perçu à l'intérieur ou très proche de la tête. Pour obtenir les meilleurs résultats, il est alors fortement conseillé d'estimer et d'inverser ce filtrage complémentaire, ce qui est très rarement fait en pratique. En outre, des études récentes [WK05] ont également montré que les différences entre casques de même marque et modèle pouvaient être significatives, typiquement du même ordre que les différences entre HRTFs de différents sujets. Chaque casque devrait donc, théoriquement, être individuellement calibré pour une restitution optimale. Ces problèmes ainsi que la dépendance des HRTFs vis-à-vis de l'auditeur expliquent sans doute le succès limité de ces techniques auprès du grand public alors que, théoriquement, elles devraient pouvoir offrir la meilleure solution possible en terme de cot d'implementation et de qualité du résultat. Les approches binaurales restent néanmoins considérées comme les approches de référence pour la spatialisation du son, en particulier lorsqu'elles peuvent être spécifiquement personnalisées pour l'auditeur. De plus, il a été démontré qu'il est possible de s'adapter à des HRTFs génériques [BKW04]. Une étape d'entranement et d'adaptation (probablement nécessaire dans tous les cas en pratique) pourrait donc au final se substituer à l'utilisation de filtres individualisés, plus délicats à obtenir.

Binaural recording

Si le filtrage d'un son monophonique par des HRTFs est une solution très courante pour la spatialisation du son, il est également possible d'acquérir des scènes sonores réelles dans un format stéréophonique propice à la restitution binaurale. De la même manière que pour la mesure des HRTFs, on peut utiliser une paire de microphones placés dans les oreilles d'un sujet ou d'une tête artificielle. On peut également utiliser une paire de microphones permettant une décomposition directionnelle du champ sonore en deux points. Cette dernière solution offre l'avantage d'obtenir un enregistrement indépendant de la morphologie de l'auditeur. L'individualisation de la restitution peut donc être réalisée en post-traitement. Un exemple d'une telle technique est le "binaural B-format" [JWL98, JLP99], utilisant des microphones Sound-field [Sou] par exemple. D'autres approches ont également proposé d'utiliser un ensemble de microphones répartis sur une sphère [ADT04] puis de les réinterpoler en fonction d'un suivi de l'orientation de la tête de l'auditeur lors de la restitution. Néanmoins, ces techniques de prise de son ne permettent pas la post-édition de la scène sonore (déplacement de l'auditeur, des sources, etc.), qui est figée dans l'enregistrement. Les techniques d'enregistrement binaurales, couplées à un système de reproduction stéréophonique par oreillettes permettent également d'envisager des applications de réalité auditive augmentée dans lesquelles la scène sonore réelle captée aux deux oreilles est enrichie par des sons virtuels [LNV+04].

Efficient implementation

L'implémentation efficace d'un dispositif de spatialisation binaurale, se base généralement sur une séparation des indices ITD, ILD et la composante spectrale des HRTFs. L'ITD et l'ILD sont alors considérés comme un retard et un gain inter-aural indépendant de la fréquence. Les HRTFs sont implémentées comme des filtres plus ou moins complexes ou une simple correction en fréquence (égalisation). Ce type de modèle permet une première possibilité de personnalisation simple des indices ITD et ILD en fonction de paramètres morphologiques de l'auditeur. Il permet également d'interpoler le spectre des HRTFs et de prendre en compte le délai de propagation et l'ILD en une seule opération de ligne-à-retard non entière qui est en général facilement réalisée dans le domaine temporel. La partie filtrage des HRTFs peut, quant-à-elle, être réalisée, en domaine temporel, par des filtres à réponse impulsionnelle infinie

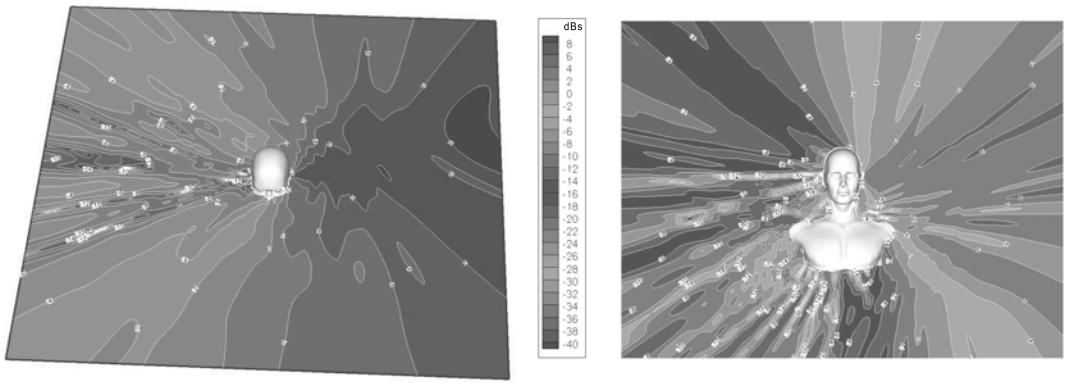


Figure 3.4: Simulation numérique des HRTFs par éléments finis de frontière. Amplitude de la pression acoustique dans un plan horizontal (à gauche) et médial (à droite) autour d'un modèle de buste humain. La source sonore est placée dans l'oreille gauche du modèle. Le calcul est réalisé pour une fréquence 8kHz et devra être reconduit pour de nombreuses autres fréquences afin de reconstruire la fonction de transfert souhaitée.

d'ordre faible ou bien, en domaine fréquentiel, par des filtres à réponse finie courts (typiquement 32 à 256 points à 44KHz). Afin de réduire la quantité d'information stockée et de permettre une interpolation plus aisée des filtres, on peut également utiliser une analyse des HRTFs en composantes principales ou indépendantes [CVH95, Lar01, VJGW00]. Ceci permet alors de ne représenter les HRTFs qu'en fonction d'un nombre limité de filtres de base. La spatialisation s'effectue alors par combinaison linéaire du signal source filtré par les différents filtres de base. Le filtrage peut être fait en post-traitement sur les k canaux de base correspondant à k copies des signaux d'entrées dont les pondérations (issues de l'analyse en composantes principales) dépendent de la direction. Cette approche permet également une optimisation des calculs dans le cas de sources sonores utilisant le même signal ou de réfiexions du son puisque le filtrage n'est effectué qu'une seule fois pour chaque source en pré-calcul et seule la combinaison linéaire varie suivant la direction de provenance du son.

Transaural techniques

Enfin, on notera qu'il est également possible de restituer le son binaural sur haut-parleurs plutôt qu'au casque grce aux techniques dites "transaurales" [Møl89, Gar95, JJ00]. Dans ce cas, il est nécessaire d'éliminer les contributions croisées des haut-parleurs, c'est à dire le son du haut-parleur droit arrivant à l'oreille gauche et vice-versa (Figure 3.5). De tels dispositifs permettent une meilleure externalisation des sources sonores mais au prix d'un positionnement très précis et fixe de la tête de l'auditeur ou de l'utilisation d'un système de tracking. Bien que des effets d'élévation puissent alors être reproduits, les effets arrière sont plus difficiles à obtenir avec de tels systèmes. En général, l'azimuth maximum possible reste limité à $\pm 90^o$. L'utilisation d'une seconde paire de haut-parleurs situés derrière l'auditeur peut alors permettre d'obtenir de meilleurs résultats.

Des ondes ultrasoniques peuvent également être utilisées pour créer des faisceaux sonores très directionnels, qui pourraient être utilisés pour la synthèse binaurale sur haut-parleurs en limitant les problèmes de contributions croisées [Pom99]. Cette technique se fonde sur les propriétés de l'air dans lequel la propagation devient non linéaire à des niveaux de pression sonore très élevés. Ainsi, il est possible de transmettre deux ondes ultrasoniques de très haute intensité, par exemple à 100 kHz et 101 kHz et pro-

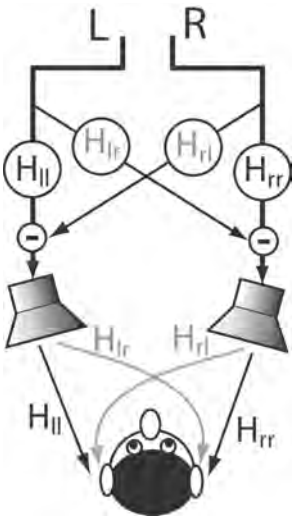


Figure 3.5: Principe de la restitution transaurale. Les contributions croisées des haut-parleurs vers les oreilles de l'auditeur doivent être estimées et éliminées. Pour cela, il est nécessaire de connaitre ou d'estimer les fonction de transfert Hlr et Hrl.

duire une onde audible à 1kHz comme résultat de leur intermodulation. Néanmoins, le signal résultant souffre d'une distorsion significative et un tel système ne permet pas la fidélité de reproduction des haut-parleurs traditionnels. En outre, il est peu approprié à la reproduction des basses fréquences.

3.2.2 Holophony and decomposition on spatial harmonics bases

Une seconde catégorie de dispositifs s'intéresse à la modélisation du champ sonore physique dans une certaine zone de l'espace autour de l'auditeur et à sa restitution. Ces dispositifs se fondent sur des formalismes mathématiques exprimant une solution à l'équation d'onde dans la zone de restitution. Parmi ces approches, on peut distinguer l'holophonie (en anglais, holophony ou wave-field synthesis (WFS)) et les approches de type "décomposition en harmoniques", comme Ambisonics, par exemple.

Holophony

Dans une représentation holophonique, le champ acoustique dans la zone de restitution est exprimé, à travers le formalisme de Kirchhoff-Helmholtz, comme la somme de sources secondaires situées sur une surface fermée quelconque entourant cette zone. La relation suivante lie la pression P0 et le gradient de pression fiP0 du champ acoustique créé par les sources primaires sur la surface S à la pression P en tout point de la zone qu'elle englobe (voir notations en Figure 3.6):

$$P(x_0,x) = \int\int_S [\nabla P_0(\mathbf{x},f).\mathbf{n} - \mathbf{R}.\mathbf{n}(1+jkr)P_0(\mathbf{x},f)/r]\,\frac{e^{-jkr}}{4\pi r}dS, \tag{3.1}$$

où x_0 est le point de restitution, x est la variable d'intégration sur la surface englobant la zone de restitution, P est la pression acoustique créée par les sources secondaires sur la surface, P_0 la pression des sources primaires, $\mathbf{n}$ est la normale unitaire à la surface en x, $k = 2\pi/c$ est le nombre d'onde, $r = ||\mathbf{R}||$ et $j^2 = -1$.

L'intérêt principal de la représentation holophonique réside dans la validité de reproduction pour une large zone de l'espace qui convient très bien à des situations multi-utilisateurs. En outre, le principe

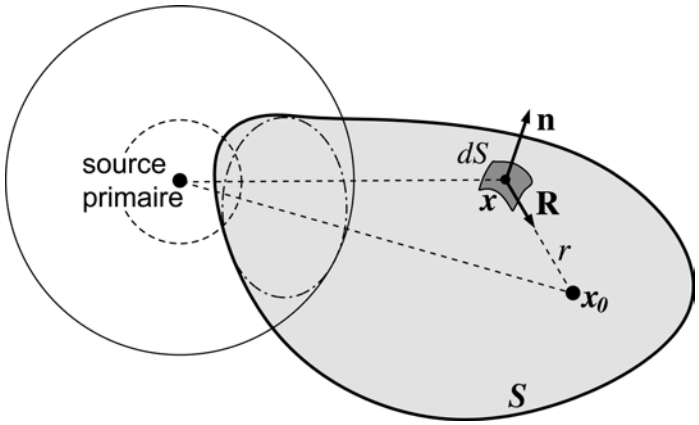


Figure 3.6: Notations pour l'intégrale de Kirchhoff-Helmholtz

d'holophonie implique directement un procédé d'enregistrement-restitution. Il est en effet possible d'enregistrer des environnements sonores réels à l'aide de multiples microphones de pression (omnidirectionnels) et de gradient de pression (à directivité en "figure de huit") répartis sur une surface (ou un contour) et les reproduire ensuite directement sur des haut-parleurs répartis sur la même surface (ou contour). Il est également envisageable de reproduire le champ sonore à l'aide d'une surface différente à condition qu'elle soit contenue dans l'espace défini par la surface d'enregistrement [PB04a].

Dans son principe et son formalisme, l'holophonie suppose la reproduction du champ sonore grce à une infinité de sources secondaires distribuées de façon continue sur une surface fermée (formalisme Kirchhoff-Helmholtz) ou un plan infini (formalisme Rayleigh) séparant le domaine des sources et le domaine de reproduction. Une mise en pratique simplifiée est due aux travaux de Berkhout et de Vries [BdVV93]. Elle consiste en un ensemble d'approximations et de termes correctifs associés. La première suppose que la majorité des sources virtuelles utiles est située dans le plan horizontal où se situent également les auditeurs. Dans ce cas, la contribution majeure des sources secondaires est due aux sources secondaires situées dans ce plan horizontal. On peut donc en première approximation et moyennant des termes correctifs, réduire la distribution surfacique de sources à une distribution linéaire. Les autres approximations consistent à réduire cette distribution linéaire à un ou plusieurs segments (ce qui nécessite des procédures de minimisation des effets de diffraction) et inévitablement à utiliser un nombre fini de haut-parleurs régulièrement espacés. Cet échantillonnage spatial s'accompagne d'une fréquence d'aliassage spatiale au-delà de laquelle l'exactitude de la reproduction ne peut plus être garantie. En pratique on choisit cette fréquence au dessus de 1kHz de sorte à assurer la fidélité de reproduction de l'indice interaural de retard de phase sur l'ensemble de la zone d'écoute.

Decomposition on harmonics bases

Les approches utilisant une décomposition en harmoniques, comme Ambisonics [Ger85, MM95, Lee98], représentent le champ sonore incident en un point (la position de l'auditeur) comme une distribution angulaire de pression sonore qui évolue au cours du temps. Cette distribution $P(\theta,\phi,t)$ peut alors être exprimée sur une base de fonctions sur l'espace des directions, comme les harmoniques sphériques [Hob55]. Les harmoniques permettent de réaliser une analyse en fréquence, similaire à une analyse de Fourier, dans

l'espace des directions. Les coefficients de la décomposition sont alors obtenus comme :

$$p^{l,m}(t) = \int_{\theta=0}^{2\pi} \int_{\phi=0}^{\pi} P(\theta,\phi,t) y_l^m(\theta,\phi) \sin\theta d\theta d\phi,$$

où P est la distribution de pression et les y_l^m sont les m termes d'harmoniques sphériques d'ordre l. Pour plus de détail sur l'expression des harmoniques sphériques, nous invitons le lecteur à consulter les références [Hob55, PB04a]. A l'inverse, la reconstruction à partir des coefficients d'harmoniques sphériques est obtenue comme :

$$P(\theta,\phi,t) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} p^{l,m}(t) y_l^m(\theta,\phi).$$

L'intérêt principal de la projection du champ sonore sur une base d'harmoniques sphérique réside dans les manipulations qui peuvent être directement effectuées par combinaison linéaire des signaux de base. Par exemple, l'ensemble du champ sonore peut être ré-orienté en 3D pour compenser, par exemple, la rotation de la tête de l'auditeur. On peut également modifier la "perspective" de la scène sonore, ou bien réaliser des interpolations directionnelles entre différents enregistrements. A l'ordre 1, seuls 4 signaux sont nécessaires pour effectuer ces opérations, ce qui rend la méthode très efficace. Toutefois, dans ce cas, la résolution spatiale reste très limitée. En plus de permettre une formalisation pour l'acquisition, le traitement et la restitution d'un champ sonore 3D, un intérêt majeur de ces techniques est qu'elles permettent une description du champ sonore indépendante du dispositif de restitution. Une même représentation, par exemple obtenue à partir d'enregistrements, peut alors être décodée sur différents dispositifs (casques, haut-parleurs, etc.). Elles offrent également la possibilité de dégrader progressivement la qualité de la restitution ou du traitement en jouant sur l'ordre de la décomposition utilisée. Un autre intérêt du formalisme à base d'harmoniques sphériques est qu'il est possible de réaliser des dispositifs de prise de son permettant de capturer et manipuler un champ sonore réel. Le microphone Soundfield (Figure 3.7), disponible commercialement, permet par exemple de réaliser ce genre d'enregistrements à l'ordre 1. Ce format d'enregistrement, nécessitant quatre canaux est communément appelé B-format. Dans ce cas, un canal correspond à une directivité omnidirectionnelle et 3 canaux correspondant à une directivité en "figure de 8" sont orientés suivant trois directions orthogonales (Figure 3.7). Une unité de traitement permet de réaliser, à partir du signal capté par les capsules, la projection du champ sonore 3D sur la base d'harmoniques sphériques. Ces signaux peuvent être également recombinés de manière à augmenter et diriger la directivité du microphone ("beamforming" en anglais), ce qui ouvre de nombreuses autres applications en prise de son. Des prototypes permettant des enregistrements à l'ordre supérieur (4 à 5) ont également été récemment réalisés, comme le Eigenmike [ME04a] (Figure 3.8). Néanmoins, ils souffrent d'une bande passante réduite, ce qui n'est pas un inconvénient majeur pour des applications de télécommunication mais compromet leur utilisation en production audio. De plus, le nombre de canaux nécessaires à l'acquisition de champs sonores sur le terrain croit rapidement avec l'ordre de la représentation, ce qui limite cette approche en pratique à des ordres faibles.

Ces formalismes à base d'harmoniques sphériques ont récemment été rapprochés d'une solution à l'équation d'onde permettant de représenter le champ sonore dans une région de manière similaire à l'holophonie. Cette représentation utilise une décomposition en série de Fourier-Bessel. L'expression de la pression acoustique peut alors être reconstruite à partir des coefficients de décomposition de Fourier-Bessel de la manière suivante :

$$P(r,\theta,\phi,t) = 4\pi \sum_{l=0}^{\infty} \sum_{m=-l}^{l} P_{l,m}(t) j^l j_l(kr) y_l^m(\theta,\phi),$$

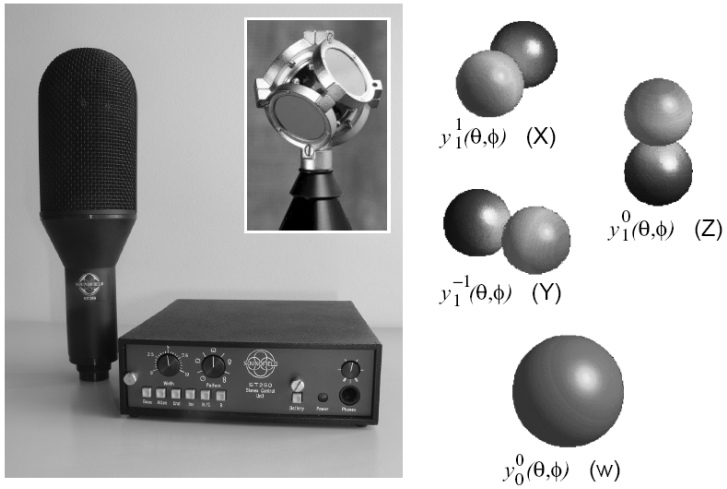


Figure 3.7: Microphone *Soundfield ST250* constitué d'une antenne de quatre capsules (voir vignette) et d'un boîtier de traitement dédié. Ce microphone permet de réaliser différentes configurations de prise de son stéréophonique de type M/S. Il permet également un enregistrement au format Ambisonics d'ordre 1, correspondant aux quatre premières composantes d'une décomposition directionnelle du champ sonore en harmoniques sphériques (à droite).

où le terme $j^l j_l(kr) y_l^m(\theta,\phi)$ est la fonction de Fourier-Bessel d'ordre l et de terme m composée de la fonction de Bessel sphérique de première espèce d'ordre l, $j^l j_l(kr)$ et de l'harmonique sphérique $y_l^m(\theta,\phi)$. Des détails complémentaires peuvent être trouvés dans [PB04a].

Dans l'hypothèse où les sources sonores sont situées à l'infini, cette décomposition rejoint la décomposition en harmoniques sphériques décrite précédemment. Cette représentation peut donc être vue comme une généralisation de la décomposition en harmoniques, qui permet de représenter également la dépendance du champ sonore en fonction de la distance et non seulement de la direction.

Là encore, des dispositifs de prise de son équivalents existent (Figure 3.9) mais souffrent des mêmes limitations sur l'ordre de décomposition que les dispositifs basés sur des harmoniques sphériques. Dans certains cas particuliers toutefois, il est possible d'optimiser le dispositif de prise de son si le dispositif de restitution est fixé. C'est le cas, par exemple, du dispositif Trinnov Surround Recording Platform [TRI] qui permet de recréer des directivités d'ordre 5 avec seulement 8 microphones ominidirectionnels dont la disposition spatiale est optimisée pour une restitution sur un système 5.1 (Figure 3.10). Ce dispositif permet en outre d'utiliser des microphones de studio classiques et donc d'obtenir une meilleure qualité d'enregistrement que des microphones utilisant un grand nombre de capsules.

Influence of spatial sampling and decomposition order

Un inconvénient des approches holophonique ou de décomposition en harmoniques est qu'elles doivent utiliser un nombre important de haut-parleurs pour reconstruire le champ sonore avec une résolution suffisante dans la zone de restitution. En effet, dans le cadre de la mise en oeuvre pratique de l'holophonie, il est nécessaire d'échantillonner la surface englobant la zone d'écoute. Pour une bonne reconstruction, il est donc nécessaire de placer les haut-parleurs de telle sorte que leur écartement respecte le critère de Shannon et soit donc deux fois plus petit que la plus courte longueur d'onde à reproduire. En pratique, cela impliquerait si l'on voulait restituer une onde jusqu'à 20kHz de placer des haut-parleurs tous les 8mm autour de la zone d'écoute, ce qui est irréalisable actuellement. Le même problème

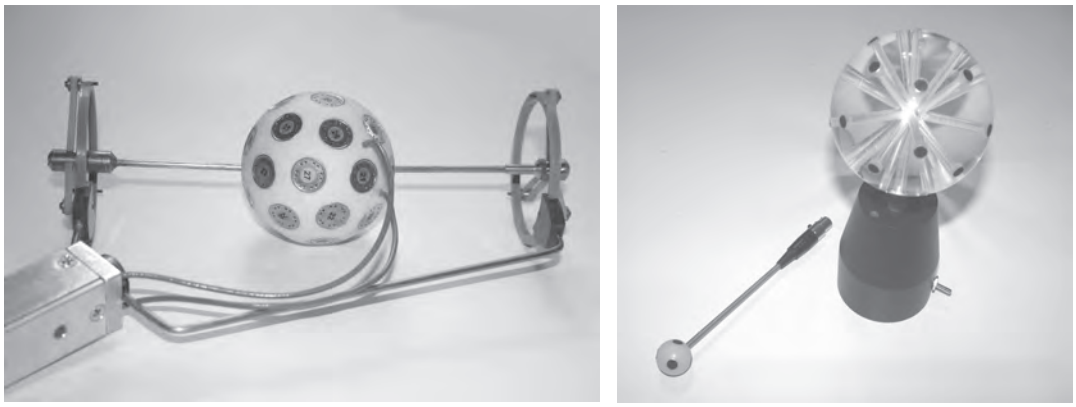


Figure 3.8: Différents prototypes de ”Eigenmicrophones” réalisés par la société Murray Hill Acoustics. A partir d'un nombre de capsules réparties sur la surface d'une sphère (32 dans le microphone de gauche, respectivement 6 et 24 dans les microphones de droite), il est possible d'obtenir une décomposition du champ sonore sur une base d'harmoniques sphériques d'ordre 4. Une application directe est le contrôle et l'orientation de la directivité du microphone. Sur ces exemples le diamètre des plus gros microphones est 6.5 cm, le plus petit 1.5 cm. MH Acoustics LLC.

Figure 3.9: Prototype de microphone à 24 capsules permettant d'enregistrer un champ sonore tridimensionnel à partir d'un formalisme en décomposition de Fourier-Bessel. Trinnov.

Figure 3.10: Trinnov Surround Recording Platform. Système de prise de son à haute résolution spatiale et processeur associé pour la capture et le traitement de scènes sonores en format 5.0. Trinnov.

existe pour les approches à base de décomposition en harmoniques, puisqu'en pratique l'ordre de la décomposition et le nombre de haut-parleurs nécessaires à la restitution est également limité. Les deux approches se distinguent néanmoins dans la manière dont la limitation de l'échantillonnage ou de l'ordre de représentation infiue sur la restitution. Dans le cadre d'approches de type Ambisonics, une troncation de la représentation entraîne la réduction de la zone de reproduction idéale lorsque la fréquence du son augmente. Toutefois, la représentation reste valide indépendamment de la fréquence. Dans le cadre de l'holophonie, la zone d'écoute idéale reste stable indépendament de la fréquence mais ce, bien entendu, jusqu'à la fréquence d'aliassage à partir de laquelle la reproduction devient incorrecte dans toute la zone. Pour une discussion plus détaillée de ces phénomènes, nous renvoyons le lecteur à la référence [PB04a].

3.2.3 Comparison and integration in virtual reality environments

Les dispositifs de spatialisation binauraux permettent une excellente qualité de restitution mais sont principalement limités à un contexte mono-utilisateur. Néanmoins, on peut les étendre à des contextes multi-utilisateur si on équipe chacun des auditeurs d'un casque (sans fil par exemple). Combiné à des systèmes de suivi de mouvement, ce genre de dispositif a été utilisé avec succès dans de nombreuses applications de réalité virtuelle et de réalité sonore augmentée. Ils permettent également une intégration aisée du son spatialisé dans des environnements de type "CAVE" dans lesquels l'intégration de haut-parleurs est plus délicate. Les systèmes de rendu multi haut-parleurs s'affranchissent de certaines limitations de la restitution binaurale, en particulier la dépendence individuelle. En outre, ils permettent également une présentation plus aisée dans un contexte multi-utilisateur. Les systèmes de type "pan-pot" offrent une spatialisation efficace dans le plan horizontal et une intégration facile puisque les dispositifs de restitution sont standardisés dans l'industrie pour le cinéma. Par ailleurs, ces dispositifs sont largement disponibles auprès du grand-public et peuvent être bon marché. De fait, ce sont les techniques de spatialisation les plus communément utilisées dans l'industrie du jeu vidéo. Elles suffisent à reproduire des effets "surround" très marqués bien que leurs performances se dégradent très vite en dehors de configurations bien maîtrisées ou calibrés. Les dispositifs basés sur des formalismes de type harmoniques sphériques et holophonie offrent le meilleur compromis entre zone d'écoute et qualité de la restitution spatiale. Ils permettent un rendu amélioré des effets de distance et parallaxe sonore mais nécessitent un grand nombre de haut-parleurs, ce qui peut rendre leur intégration difficile. De fait, les systèmes holophoniques sont souvent limités à une "fenêtre" acoustique devant l'auditeur, par exemple accompagnant un rendu visuel sur grand écran. Les techniques ambisoniques nécessitent, sinon

Figure 3.11: Différentes installations de réalité sonore augmentée réalisées lors du projet LISTEN. A gauche : installation artistique Raumfaltung - Beat Zderer/R.G.Arroyo/G.Eckel. A droite : installation didactique - Macke Labor. Photos Friedhelm Schulz - Kunstmuseum Bonn

une répartition spatiale régulière, une répartion qui entoure complètement l'auditoire car l'ensemble des haut-parleurs, y compris à l'arrière, est sollicité dans le cas de la reproduction d'une source frontale. Les Figures 3.11,3.12 et 3.13 montrent quelques exemples d'intégration de systèmes de rendu sonore spatialisé dans des systèmes de réalité virtuelle ou augmentée. L'intégration de la modalité auditive couplée à un dispositif de tracking permet d'envisager de nouvelles formes d'applications multimédia basées sur un principe de réalité augmentée. Le projet européen LISTEN (IST-1999-20646) réalisé en 2003 fournit une illustration de l'application de la notion de réalité sonore augmentée dans un contexte muséographique. Sur le plan technique, les visiteurs du musée étaient dotés de casque sans fil et de dispositifs de captation de position permettant d'asservir le contenu audio présenté à leur position et parcours dans l'exposition. Dans LISTEN, l'utilisateur parcourt un lieu réel auquel est superposé un univers sonore constitué de divers éléments, motifs musicaux ou messages didactiques, répartis dans l'espace et éventuellement associés à la présence physique d'un objet (un tableau, une sculpture). Comme illustré sur la Figure 3.11, l'interaction est fondée sur une tche de navigation de l'auditeur pour explorer l'organisation spatiale des éléments sonores. La perception sonore spatiale peut reposer sur un modèle réaliste de la scène respectant les lois de l'acoustique physique. L'interaction peut également être enrichie en insérant des règles de partitionnement de l'espace (zones de déclenchement d'événements) et de dépendance temporelle (analyse du parcours de l'auditeur), incitant dès lors l'auditeur à jouer avec le contenu et tenter de l'influencer par son comportement.

3.2.4 Latency and synchronization with other modalities

Un facteur important pour les applications immersives impliquant le son en complément d'autres modalités (visuelle, haptique en particulier) est la latence globale des différents systèmes de rendu et leur synchronisation. Différentes études ont été conduites pour estimer l'infiuence de la latence globale d'un système de rendu sonore avec suivi de la tête sur la précision de localisation des sources virtuelles. Ainsi des travaux récents [Wen01] [Wenzel, 1999, Miller e.a., 2003] montrent qu'une latence de l'ordre de 500ms ne détériore pas la qualité de localisation. En outre, en dessous de 250ms la latence du suivi de tête sur le rendu auditif n'est pas perçue de manière significative par les sujets. Si la localisation auditive semble peu affectée par la latence, il n'en est pas de même pour le suivi d'évènements multi-modaux synchrones. Dans le cas d'une simulation haptique augmentée par l'audio (par exemple, synthèse des bruits de contact), des sensibilités moyennes à l'asynchronie de l'ordre de 20ms ont été mesurées, avec

Figure 3.12: Différents bancs de haut-parleurs pour la restitution en mode Wave Field Synthesis. A gauche : banc de haut-parleurs conventionnels. A droite : banc de haut-parleurs MAP (multi-actuator-pannels). IRCAM - Photo : Terence Caulkins

Figure 3.13: La salle immersive du Centre Scientifique et Technique du Batiment (CSTB) à Sophia Antipolis. Cette salle, de type "Reality Center", combine un système de projection stéréographique sur grand écran cylindrique, un système de reproduction sonore sur haut-parleurs "5.1" ainsi que des sièges individuellement équipés d'un système de restitution transaural (à droite). Les deux systèmes audio peuvent être utilisés simultanément. CSTB/Photo : Vincent Bourdon.

des minima de l'ordre de quelques millisecondes [Adelstein e.a., 2003]. Pour une brève vue d'ensemble de ces phénomènes on pourra également se reporter à [Levitin e.a., 2000].

3.3 Software libraries for spatial audio rendering

Certaines des techniques de restitution sonore spatiale décrites dans ce chapitre sont implémentées dans diverses librairies logicielles publiquement disponibles qui peuvent constituer un bon point de départ pour l'intégration de son spatialisé dans une application de réalité virtuelle. Ainsi, *Direct Sound* [Dir04], le composant audio de la librairie DirectX de Microsoft fournit un support pour la spatialisation du son sur enceintes ou au casque. L'intérêt de cette librairie est l'accélération matérielle de ses fonctionnalités, supportée par une vaste majorité des cartes sons (en particulier grand public). Les spécifications de la librairie n'imposent toutefois pas d'algorithme pré-défini pour réaliser la spatialisation du son proprement dite, dont la qualité est donc dépendante du vendeur. De plus, cette librairie est limitée à un unique type de plate-forme, MS Windows. Pour répondre à cette limitation, une alternative, *OpenAL* [OPE00], a été développée principalement par le constructeur de cartes son Creative [Sou04]. OpenAL reprend et complémente les fonctionnalités de DirectSound et propose une interface indépendante de la plate-forme et moins lourde à utiliser. Néanmoins, la encore, le support matériel est en pratique limité aux plate-formes MS Windows et aux cartes Creative qui est reste le principal acteur de son développement. Ces deux librairies sont disponibles gratuitement. Il faut également noter que la plupart des cartes sons grand public du marché supportent au niveau matériel le rendu d'effets environnementaux (occultations, réverbérations, etc.) à travers la norme *IASIG* 2.0 [IAS], qui se base sur le système EAX [EAX04] créé par Creative. EAX, qui a évolué depuis plusieurs années, comprend également dans sa version actuelle (5.0) des effets modulaires et un meilleur traitement des occultations et transmissions mais n'est accéléré que sur les cartes Creative.

FMOD [FMO] implémente également des fonctionnalités similaires aux précédentes mais offre un support logiciel optimisé et de bonne qualité sur les plate-formes ne disposant pas d'accélération matérielle, ce qui n'est pas le cas d'openAL ou Direct-Sound à l'heure actuelle. Cette librairie est également disponible gratuitement pour des applications non commerciales. Une alternative commerciale est la librairie GameCoda de Sensaura/Creative [Sen01].

Pour des applications immersives, intégrant un système de tracking par exemple, le centre de recherche NASA AMES propose également une librairie logicielle de spatialisation du son en libre téléchargement :*SLAB* [Mil01, MW02, Lab] Pour des applications nécessitant une restitution de haute qualité ou une programmabilité plus importante, des solutions commerciales comme les systèmes TuckerDavis [tdt] sont également disponibles. Elles consistent principalement en une station de travail ou rack muni d'entrée/sortie audio de qualité et dont le processeur peut être programmé assez librement grace à une API ou une interface de programmation graphique dédiée. Néanmoins l'écart de performance, fonctionnalités et qualité entre ces solutions et des systèmes grand-public tend de plus en plus à se réduire. Spat, le Spatialisateur de l'Ircam, est une bibliothèque de modules de spatialisation dédiés aux applications musicales et interactives développées pour l'environnement de traitement du signal temps-réel MAX/MSP. Le Spatcouvre l'ensemble des dispositifs de reproduction, du contexte de laboratoire ou domestique (reproduction binaurale sur casque d'écoute, systèmes stéréophoniques ou 5.1) aux situations de concert (distribution 2D ou 3D de haut-parleurs) ou aux installations sonores holophoniques et interactives. L'utilisateur peut associer à chaque événement sonore de la partition des données de localisation dans l'espace ainsi qu'une description de l'effet de salle sous forme de facteurs perceptifs. Le Spatpeut être contrôlé à partir d'un séquenceur, d'un système de suivi de partition, ou à partir de processus de contrôle de haut-niveau dédiés à l'écriture musicale ou à l'écriture de scénarios interactifs (Open-Music,

Virtools, Avango, ListenSpace,...) [spa, JW95]. Enfin, des travaux récents ont également démontré la possibilité d'utiliser l'accélération matérielle des cartes graphiques programmables (GPU) pour réaliser des calculs performants de traitement du signal (transformée de Fourier rapide par exemple [BFH+04]) et spatialisation du son [GT04]. Cette approche peut être particulièrement interessante si le GPU n'est pas surchargé par l'affichage puisqu'il peut être largement plus performant que le processeur central pour des applications massivement parallèles. En outre, le hardware graphique évolue de manière beaucoup plus rapide que le hardware sonore et offre des fonctionnalités et interfaces de programmation beaucoup plus standardisées, tout en étant largement disponible sur une variété de plate-formes. Il constitue donc une alternative à explorer pour l'implémentation de traitements audio complexes.

Chapter 4

Interactive physically-based auralization

In this chapter, we introduce three contributions to interactive physically-based auralization. First, we show how interactive geometrical simulations using beam-tracing approaches can be extended to support efficient edge diffraction. We then propose a framework based on the Kirchhoff approximation that can be used to extend geometrical approaches to very complex geometry, for instance as obtained through scanning techniques. Finally, we propose a study of massively parallel architectures, such as modern graphics hardware, for audio processing and interactive acoustics simulations. We show that such architectures can be used for spatial audio rendering of massive number of sources but can also be used to accelerate scattering calculations, in particular using the Kirchhoff approximation.
More details can be found in the related publications:

- Nicolas Tsingos, Thomas Funkhouser, Addy Ngan and Ingrid Carlbom.
  Modeling Acoustics in Virtual Environments Using the Uniform Theory of Diffraction.
  Proceedings of ACM SIGGRAPH 2001.

- Nicolas Tsingos,Ingrid Carlbom, Gary Elko, Tom Funkhouser and Bob Kubli.
  Validation of Acoustical Simulations in the *Bell Labs Box*.
  IEEE CG&A, special issue on Virtual World, Real Sound, July-August 2002.

- Thomas Funkhouser, Nicolas Tsingos, Ingrid Carlbom, Gary Elko, Mohan Sondhi, James E. West, Gopal Pingali, Patrick Min and Addy Ngan.
  A Beam Tracing Method for Interactive Architectural Acoustics.
  The Journal of the Acoustical Society of America (JASA), 115(2), pp. 739-756, February 2004.

- Nicolas Tsingos, Carsten Dachsbacher, Sylvain Lefebvre and Matteo Dellepiane.
  Instant Sound Scattering.
  Rendering Techniques (Proceedings of the Eurographics Symposium on Rendering) , 2007.

4.1 Extending interactive geometrical simulations with edge-diffraction

Geometrical acoustics (GA) is one of the most widespread approaches in architectural acoustic modeling. GA is a high-frequency approximation that models sound propagation along ray-paths that specularly reflect off surfaces. The propagation paths can be constructed using techniques such as ray or beam tracing [LSVA07, FCE+98] using spatial data structures (e.g., BSP trees) to maximize efficiency. Unfortunately, they fail to simulate sound realistically because they do not accurately account for diffraction. In particular, this can result in audible discontinuities in the simulated sound when the source or

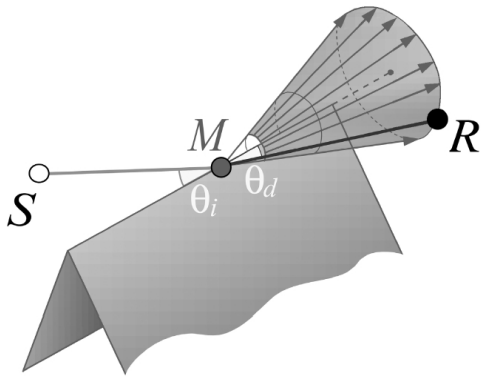


Figure 4.1: Following the uniform theory of diffraction, a ray, ρ, incident on a wedge spawns a cone of diffracted rays. The aperture angle of the cone is equal to the angle θ_i between the incident ray and the direction of the edge (the axis of the cone). For a given listening position, only one ray carries the diffracted contribution.

a strong reflection path becomes occluded from the receiver. Our first contribution is an approach to compute early geometrical propagation paths, including diffraction effects, that can be used for real-time auralization in an interactive GA system.

4.1.1 Geometrical theory of diffraction

The Uniform Theory of Diffraction (UTD) [Kel62, KP74, Han81, MPM90] incorporates diffraction into the ray theory of light. The UTD treats an infinite wedge as a secondary source of diffracted waves that in turn can be reflected and diffracted before reaching the receiver. For a given point source and point receiver location, the diffraction of a wave over an infinite wedge is represented by a single ray whose contribution to the wave field is attenuated by a complex valued diffraction coefficient. For any sequence of diffracting edges, the ray follows the path satisfying Fermat's principle: if the propagation medium is homogeneous, the ray follows the *shortest path* from the source to the receiver, stabbing the diffracting edges (Figure 4.3 left). The UTD is a high frequency approximation and applies to infinite wedges, when the source and listener remain far from diffracting surfaces (compared to the wavelength). To date, the UTD has been applied successfully in several types of off-line simulations, including acoustical diffraction over solitary wedges [Kaw81], lighting effects in room-sized scenes [AM99], and radio frequency propagation in buildings and cities [RNFR96, KGW$^+$99]. For acoustic waves, the method has been validated down to 150Hz for a small combination of diffracting wedges [Kaw81, SFV99].

4.1.2 Diffracted beams and paths

The main computational challenge in using the UTD into realtime auralization systems is the efficient enumeration of significant early diffraction paths. Although many algorithms exist to find approximate solutions [Mit98], they are either too inefficient or prone to aliasing. Our approach is based on object-precision beam tracing [HH84]. The motivation for this approach is to exploit the spatial coherence in propagation paths while avoiding the aliasing artifacts of sampling diffraction edges. In contrast to ray tracing, beam tracing works with object-precision polyhedral volumes that support well-defined intersections with diffracting edges. Aliasing resulting from the intersection of infinitely thin rays with infinitely thin edges is thus eliminated [RNFR96]. In contrast to brute-force enumeration of all edge

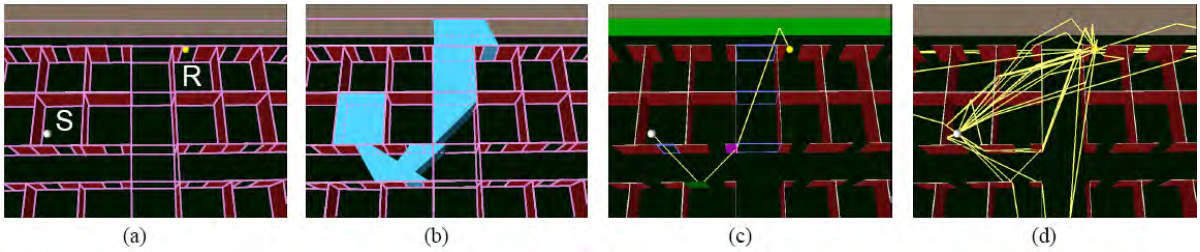


Figure 4.2: Overview of our process: (a) Virtual environment (office cubicles) with source S, receiver R, and spatial subdivision marked in pink. (b) Sample reflected and diffracted beam (cyan) containing the receiver. (c) Path generated for the corresponding sequence of faces (green), portals (purple), wedges (magenta). (d) The procedure repeated for all beams containing R.

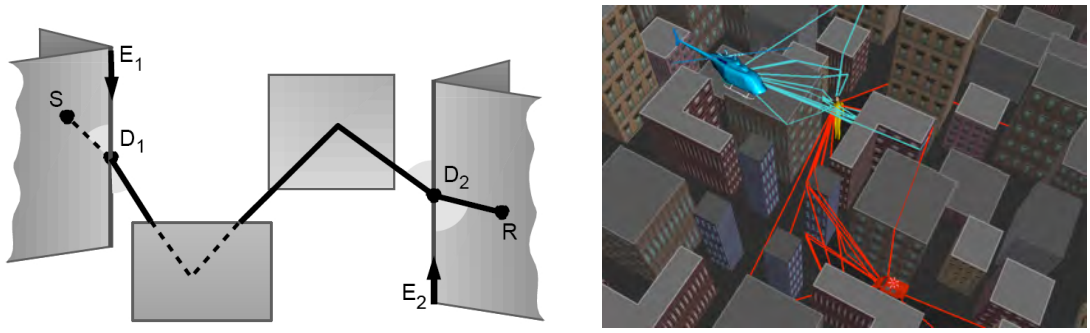


Figure 4.3: Left: A single propagation path comprising a diffraction, two specular reflections, and another diffraction. The two diffraction points (P_i) are determined by equal angle constraints at the corresponding edges (E_i). Right: Early diffracted and reflected sound paths in a city environment where direct sound from sources is occluded.

permutations [AM99], beam tracing provides an effective method for pruning the search based on the feasibility of stabbing lines [Tel92]. As a result, beam tracing finds every propagation path up to a specified termination criteria without undersampling errors. Moreover, beam tracing algorithms are practical for specular reflection in densely occluded virtual environments [FCE+98], and can be readily incorporated into interactive virtual environments systems [FMC99]. Once possible propagation sequences have been identified by beam tracing, the corresponding propagation paths are built by solving an equation system enforcing the equal angle constraints on the diffraction edges. Figure 4.2 gives an overview of our approach.

We constructed a simple test environment, the *Bell-Labs Box* to verify the validity of our combined specular reflection and edge diffraction approach. Using measured impulse response for the sound source and surfaces, we were able to achieve simulations closely matching the direct measurements, as can be seen in Figure 4.4.

Shadow-region approximation for edge diffraction

Our beam tracing technique provides a method for finding diffraction paths efficiently and without aliasing. However, contrary to specular reflections or transmissions, diffraction introduces a scattering in all directions around the wedge, which results in a combinatorial explosion of the number of beams to consider, even for moderately complex scenes. We also introduce an approximation to reduce the

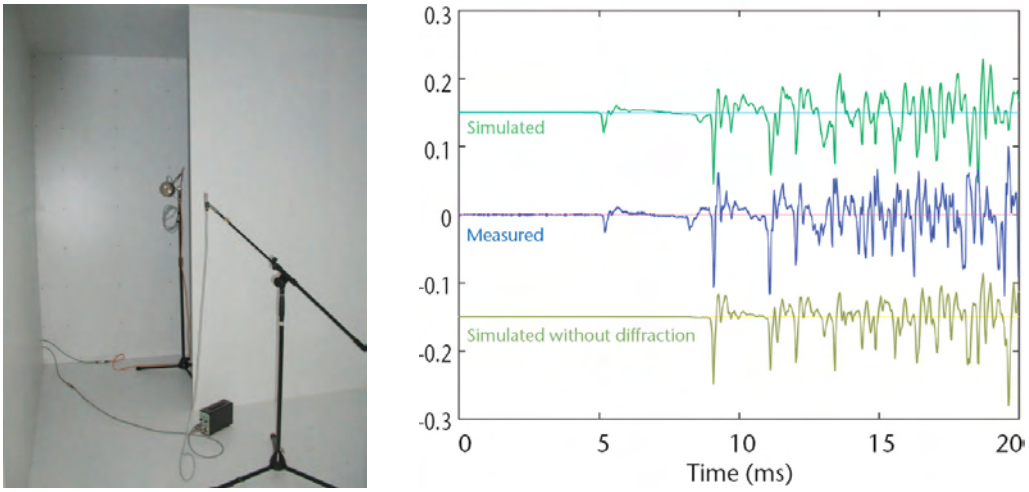


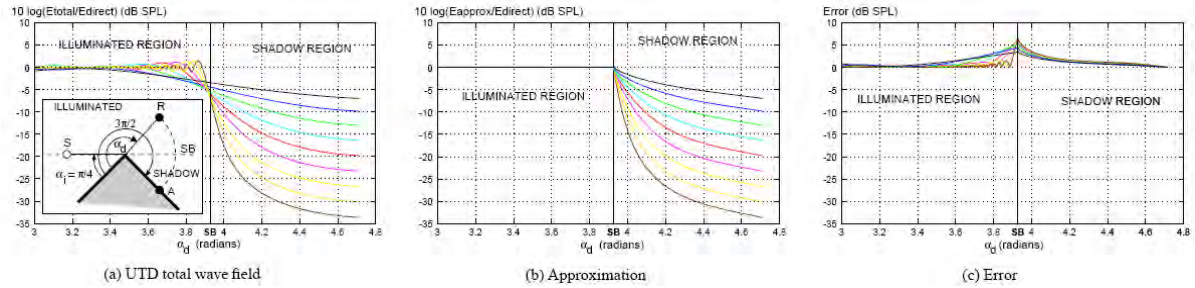
Figure 4.4: Left : A simple enclosure, the *Bell-Labs Box* was constructed to evaluate. We can mount additional panels inside the enclosure Box to study the effects of sound diffraction. Right: Comparison of a simulated early impulse response (top) including the first two orders of diffraction from the edge of the panel and the first four orders of specular reflection and a measured response (middle) in the Bell Labs Box with a baffle. The simulation computed the contribution of 1,358 propagation paths. The bottom plot shows a simulation including the first eight orders of specular reflection but omitting diffraction (307 paths).

spatial extent of diffraction beams, while preserving a good modeling of the diffracted field. We conjecture that diffraction does not modify the main acoustic cues already carried by the direct and reflected contributions since the amplitude of the diffracted field is usually much weaker than direct or even reflected contributions [Pie84] (p.500). On the other hand, we note that diffraction into shadow regions is crucial for typical virtual worlds as it provides the primary mode of propagation to most of the environment. Thus, we introduce an approximation in which the contribution of diffraction is considered only in shadow regions. This approximation enables us to achieve interactive auralization in large environments. Figure 4.5 illustrates this approximation on a single wedge example. Using this approximation, our approach can be applied at interactive rates for dynamic auralization of sound sources in virtual environments as illustrated in Figure 4.3 (right).

4.2 Beyond edge diffraction: scattering from highly detailed models

As we described in the previous section, interactive GA simulations can be enhanced by introducing diffraction effects from wedges. However, as all GA models, the geometrical theory of diffraction (GTD) assumes edges to be large compared to the wavelength. Increasing geometrical complexity would imply using smaller primitives and eventually would fall outside the validity domain of GA. Thus, it is unclear how classical GA+GTD approaches could apply to more detailed scenes. Recent works have been devoted to level-of-detail (LOD) approaches for GA [JMT03, WRR04, Sil05] but to our knowledge no general simplification scheme that preserves the correct scattering properties has been proposed to date.

Our second contribution aims at computing scattering from very complex geometry, such as highly tesselated CAD-CAM models or those acquired through scanning techniques using an efficient implementation of the Kirchhoff approximation. A key aspect of the Kirchhoff approximation is that it can be



(a) UTD total wave field (b) Approximation (c) Error

Figure 4.5: Plots of (a) the UTD total wave field, (b) our approximation, and (c) the error as a function of diffraction angle (α_d), as the receiver rotates around the edge, for a single diffracting wedge (inset). Each plot shows several curves corresponding to the sound pressure level (SPL) for the center frequencies of octave bands ranging from 100 Hz (top) to 24kHz (bottom). Our approximation culls the diffracted field contribution in the illuminated region of the wedge but still closely matches the original UTD field.

efficiently implemented on modern graphics hardware, which we describe in Section 4.3

4.2.1 Boundary Element Methods and the Kirchhoff approximation

Finite element methods (FEM) are numerical solutions to the wave (Helmholtz) equation and associated boundary conditions. They are classically solved in frequency domain by subdividing the environment into small elements (voxels) but alternative time-domain formulations can also be used [SRT94]. Of special interest is the Green surface integral formulation. Using this formulation, the pressure solution $P(R)$ to the Helmholtz equation can be expressed using an arbitrary surface Σ surrounding the receiver R [FHLB99]:

$$P(R) = P_0(R) - \int_\Sigma (P(U)\nabla G(U,R) - G(U,R)\nabla P(U)) \cdot \mathbf{d}S, \tag{4.1}$$

where $\mathbf{d}S = \mathbf{n}dS$ ($\mathbf{n}$ unit vector) and $G(U,R) = -e^{ikr}/4\pi r$ is the Green function corresponding to the propagation of a spherical wavefront in free-field (see Figure 4.6).
$P_0(R)$ is the free-field pressure emitted by the source and the integral term, often called *diffracted field*, is the solution of the homogeneous Helmholtz equation associated with the boundary conditions on the surfaces of the environment.

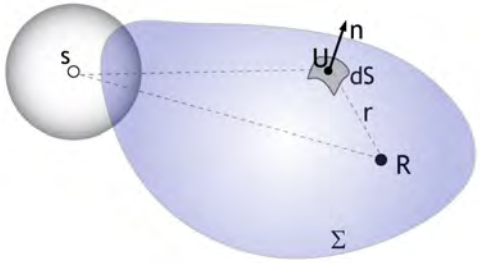


Figure 4.6: Notations for the Kirchhoff-Helmholtz integral theorem. S and R denote the source resp. receiver.

This surface integral is also called the Helmholtz-Kirchhoff integral theorem. The Green surface formulation serves as a basis for boundary element methods (BEM) techniques. In this case, Σ corresponds to the surfaces of the environment. BEM methods can account for full scattering effects in a unified way

and are widely used to compute reference solutions. However, edge-length for surface elements must typically be smaller than 1/4th of the wavelength. This makes such approaches very time-consuming at high frequencies or for large-scale problems. They also require carefully-designed meshes with uniform elements to limit errors in the solution. They are thus hard to use with most 3D models, particularly those acquired through scanning.

The Kirchhoff approximation (KA) can be seen as a hybrid strategy between GA and wave acoustics [FHLB99]. It is based on Eq. 4.1 but imposes $P(U) = P_o(U)$ and $\nabla P(U) = \nabla P_o(U)$ on the "illuminated" side of the surfaces (visible to the source) and $P(U) = \nabla P(U) = 0$ on the "shadowed" side. As a result, $P(R)$ can be computed by a direct integration but surfaces facing away from the sound source will not contribute to the solution. Neglecting the contribution from occluded surfaces and higher-order scattering is the major source of error in the KA. As a result, the approximation will degrade at very low frequencies as the scattered component becomes more important in the regions occluded from the source. The KA will also lead to inaccurate results when second-order occlusions/reflections become prominent. Several studies have shown that it can introduce significant errors in the computed scattering from simple flat or randomly-rough surfaces when compared to reference solutions [JM82, Tho87, NNK93, CL93]. In particular, errors were found to be more important at grazing angles and near-field from the surface. However, some of these studies also used additional far field approximations which might also contribute to the observed errors. As stated in [Tho87, EDS01], further work is still required to evaluate the validity of the KA which is still not well established even today. Despite these limitations, the KA has been widely used to solve scattering/occlusion problems in acoustics [SN81, CL93, CI90, TG98, Emb00, EDS01]. In [SN81] it was shown that the KA can be used to simulate impulse response of first-order reflection/diffraction off rigid panels with good agreement to measured responses.

4.2.2 Scattering from complex surfaces using the Kirchhoff approximation

We introduce a framework that can be used to approximate first-order scattering effects off arbitrary complex surfaces and pre-compute corresponding filters. These filters can be re-used within classical GA simulators. Our approach also brings an alternative solution to the problem of surface-simplification. To use the KA for complex architectural acoustics problems, we evaluate the surface integral on all geometry visible from the sound source. This integral can be computed in software using ray-casting but it also maps very well to modern graphics hardware leading to a very efficient implemention.

To evaluate our approach, we computed scattering filters for large-scale real-world situations and compared them directly to corresponding recordings taken in the field. Example audio files can be found at: *http://www-sop.inria.fr/reves/projects/InstantScattering*. A first interesting example is the Kukulkan temple, a Maya staircase-pyramid located in Chichén Itzá, Mexico (Figure 4.7 left). The stairs of this pyramid act as a sound diffraction grating. They reflect a particular chirped echo which has been the object of a number of studies [DDBL04, Bil06]. For additional information, please see *http://www.ocasa.org/MayanPyramid.htm*. Figure 4.7 (right) compares the result of our simulation to the recording. Although the recording contains significant environmental noise, the comparison shows that our algorithm convincingly captures the chirped echo from the stairs.

Our second, more challenging example models the scattering off the faade of the Duomo on the Piazza dei Miracoli in Pisa, Italy, also famous for its leaning bell-tower. We used a detailed model of the cathedral obtained from time-of-flight laser scanning (Figure 4.8 left) and containing 13 million triangles (a resolution of about 2 cm). Figure 4.8 (right) compares a simulation with an on-site recording of a handclap. The approach gives satisfying results although some components are missing, probably due to higher-order scattering or reflections from the ground (which was not acquired).

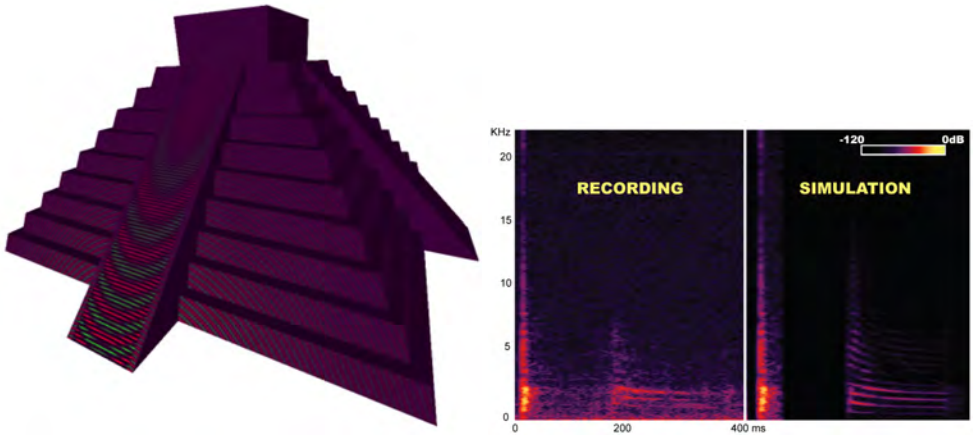


Figure 4.7: Left: Visualization of the scattering terms on the surface of a model of the Kukulkan temple for a 500Hz wave. The sound source is 15 meters in front of the stairs. Right: Comparison between spectrograms of a simulation and an on-site recording for the Kukulkan temple. The simulated response is convolved by the handclap of the original recording.

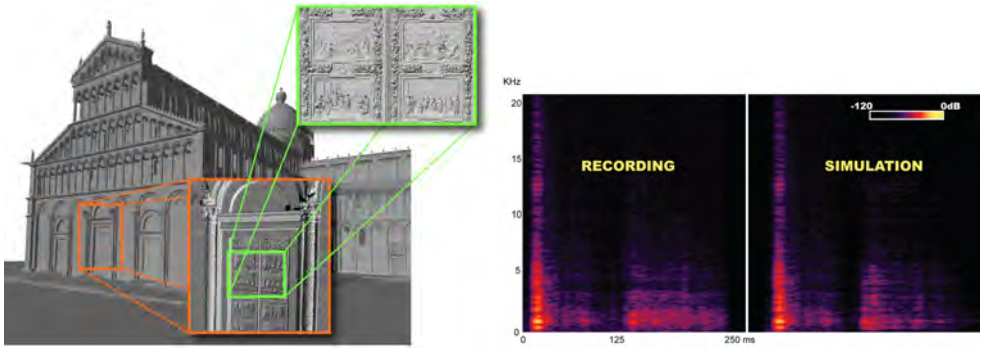


Figure 4.8: Left: A 3D model of the scanned faade of the Duomo in Pisa, Italy and close-ups on surface detail. Right: Comparison between spectrograms of a simulation and an on-site recording. The simulated response is convolved by the handclap of the original recording.

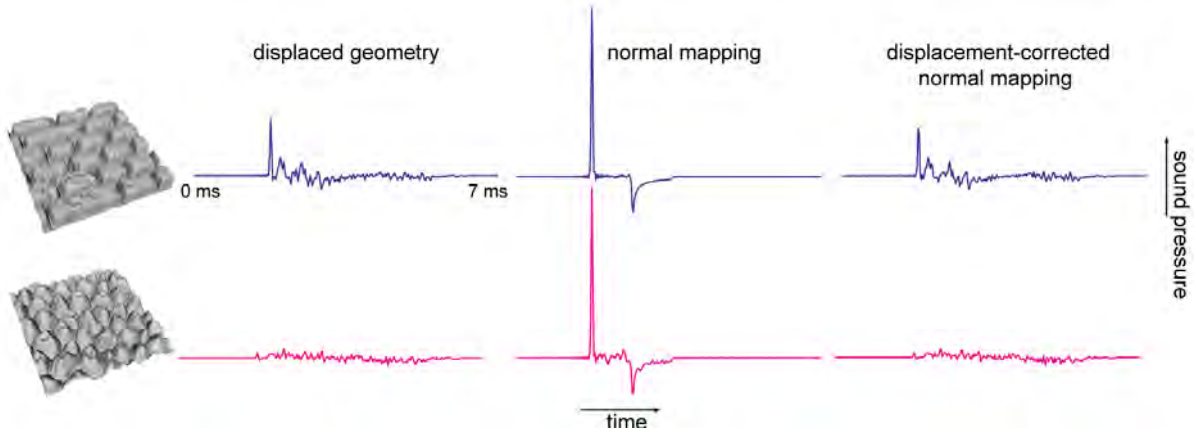


Figure 4.9: Comparison of true displaced geometry with a proxy flat quadrilateral enhanced with normal-map only or combined normal/displacement maps. Source and receiver are respectively 10 and 20 m directly above the center of the face. Note how the normal-map alone has little effect on the obtained response.

4.2.3 Integration with geometrical acoustics engines

In this section, we propose two solutions in order to integrate our KA-based approach into current GA simulators. First, the approach can be used to simplify complex geometry by introducing details as textures on a flat approximate geometry, as used in graphics rendering. Second, scattering filters from complex geometry can be pre-computed and stored in order to be directly used within classical GA frameworks.

Scatter-preserving geometrical simplification

The surface integral approach of the KA offers the possibility to leverage level-of-detail schemes originally developed for graphics rendering [COM98]. In this section, we propose a strategy combining normal mapping with displacement correction to model complex surface detail for acoustic scattering calculations. Displacement surfaces [CCC87] use textures to encode fine-grain surface detail which can be used at rendering time by a software ray-tracer or with the graphics hardware. To avoid costly ray-intersections with complex geometry, we propose a simpler approach that accounts for the correct propagation delay but omits accurate visibility calculations. To remove the contribution of back-facing fragments, which should not be contributing to the integral, we use an additional weighting term defined as $-\mathbf{v}.\mathbf{n}$, where $\mathbf{v}$ is the direction from the 3D location to the sound source and $\mathbf{n}$ is the surface normal. To evaluate our approach, we created several 4×4 meter surface samples from displacement textures. The amplitude of displacement was 0.5 meters. Figure 4.9 shows example surfaces and scattering impulse responses calculated with true displaced geometry. In this case, source and microphone were directly above the center of the surface respectively 10 and 20 meters away. We also created corresponding normal maps from the displaced geometry and performed a scattering calculation using a flat proxy surface with our displacement-corrected normal mapping and standard normal mapping only. As can be seen, normal maps without displacement correction result in very little difference compared to a flat surface. This demonstrates the importance of the interference phenomena which are paramount in modeling the proper scattering effect. Our displacement-corrected normal mapping results in a much better approximation.

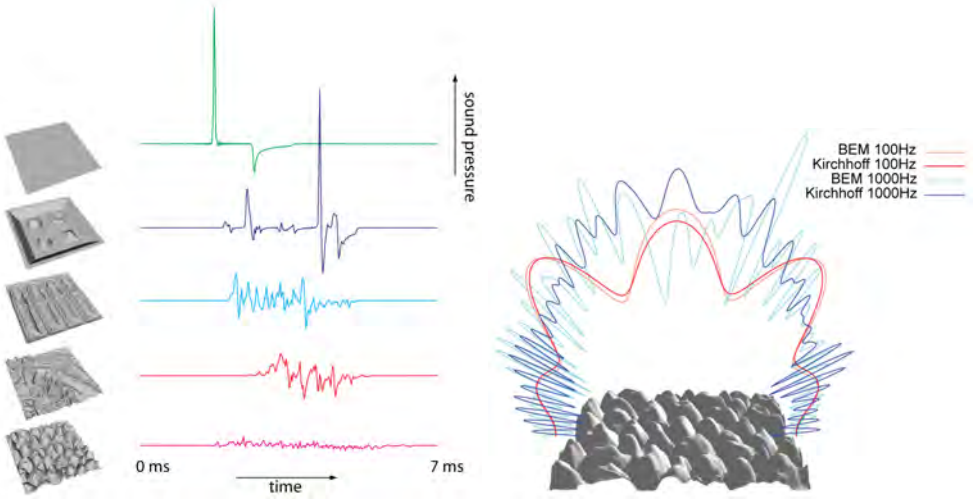


Figure 4.10: Left: Responses from different 4×4m surface samples. Each surface is composed of 131072 triangles and generated from displacement maps. Note the secondary scattering component due to the finite extent of the flat surface on the top row (green curve) and the increasingly diffusing nature of the surfaces from top to bottom. Right: Scattering patterns for a detailed surface. The figure compares sound pressure levels in a plane medial to the surface obtained by BEM and our approximation. Source is 5m directly above the center of the face and the pressure is plotted at a distance of 10m.

Using complex scattering filters with GA engines

We believe our approach can be used to enhance GA simulations with realistic pre-computed surface scattering functions or filters similar to those shown in Figures 4.9 and 4.10. For instance, the filters could be convolved along the propagation paths obtained with an image-source/beam-tracing technique. However, such an approach would fail at modeling energy transfers in non-specular directions. Hence, the approach would probably better suit a radiosity framework to account for more diffuse transfer between surfaces. In this context, our approach could also be used to compute the impulse response of the form-factors when obstacles are present between surface patches.

4.3 Using massively parallel architectures for simulation and auralization

Audio processing applications are among the most compute-intensive and often rely on additional DSP resources for real-time performance. However, programmable audio DSPs are in general only available to product developers. Professional audio boards with multiple DSPs usually support specific effects and products, while consumer “game-audio” hardware still only implements fixed-function pipelines which evolve at a rather slow pace.

The widespread availability and increasing processing power of programmable graphics hardware (GPUs) could offer an alternative solution. GPU features, such as multiply-accumulate instructions or multiple SIMD execution units, are similar to those of most DSPs [EB00]. Moreover, their high-level programmability with floating point support and easy access to development kits turns them into attractive co-processors for non-graphics applications. Besides, 3D audio rendering applications require a significant number of geometric calculations, which are a perfect fit for the GPU. In this section, we investigate the use of GPUs for efficient audio processing.

4.3.1 GPU Architecture

Graphics hardware has a specific dataflow computational model. Its architecture is originally dedicated to manipulate 3D primitives like points, lines or polygons, perform some graphics operations and render the result on the screen. Primitives follow a sequence of operations before processed to the screen. Figure 4.11 shows the essential steps of the pipeline. Basically, the application transmits data vertices of the primitives to the vertex processor. The vertex is a structure containing 3D and texture coordinates, color and normal vector. The vertex processor applies any mathematical transform to each vertex including transform from world space to projection space.

In the second step, vertices are assembled following the geometric primitives information. In this step, culling is computed to discard invisible polygons according to the normal of the polygon and the view direction. Next, clipping to the view frustum is applied before the rasterization. The view frustum is a set of plane which defined the field of view of the camera.

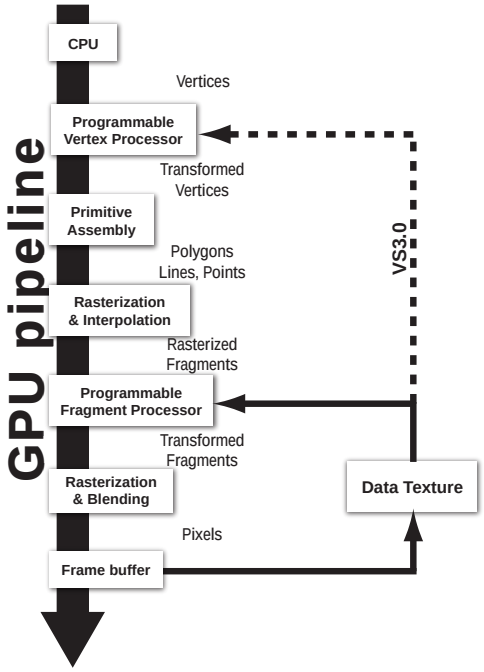


Figure 4.11: GPU pipeline. The vertex processor and the fragment processor are totally programmable.

The third step of the pipeline rasterizes the transformed primitives. Rasterization determines which fragment of the screen buffer is covered by a primitive. The term “fragment” is employed instead of pixel to make the difference between the resulting pixel and the one with other characteristics than color and containing possible operations to be done to get the result. The characteristics of the fragment, like color or textures coordinates, are interpolated between the transformed vertices of the primitive. The interpolation can be “nearest”, meaning truncated from the nearest pixel, linear or bi-linear but any other interpolations can be applied via the programming features of more recent GPUs.

The next step is the most important one. Indeed, this stage processes the operations of the fragment. Many math operations can be computed to obtain the final pixel. The aim of the last step is to perform additional tests before writing the final fragment value (color) to the frame buffer. Tests includes depth testing, used to remove hidden pixel and alpha test, for compositing purposes.

The second and the fourth step can be overridden to define user programs. It is possible to program the GPU with high level C-like languages (e.g., CG, GLSL, HLSL). The power of the GPU is provided by

its data parallel architecture. Each fragment programs is processed in parallel. G70 Nvidia card contains 24 pixels pipeline and the G80 contains 128 stream processors which are automatically attributed to vertex of fragment processing using massive multithread handling. The communication from the GPU towards the CPU is slow, even with the new PCI express bus. The fastest way to transfer resulting data back to fragment processor is to do multiple passes by rendering the frame buffer to a texture. The vertex shader 3.0 model allows transfer of data to the vertex processor. However, this communication is not really fast because all pixels have to be rendered to the frame buffer beforehand.

4.3.2 GPU-Accelerated Scattering Calculations

Evaluating the scattering approach presented in Section 4.2) (i.e., Eq. 4.1 with the Kirchhoff approximation) involves two subproblems. First, the integration domain, i.e. all the scattering surfaces visible from the source, must be determined and sampled. In itself, this is a difficult task for complex geometries. Second, the differential contribution of blocked plus reflected wavefronts must be evaluated for all surface samples and summed-up. These two tasks perfectly match the operations implemented by the graphics hardware. To maximize efficiency we implemented these two steps using a "source-view" strategy which provides the most natural mapping to the GPU architecture and results in a very straightforward implementation.

By rendering the scene from the location of the sound source, in a way similar to a *shadow mapping* technique in computer graphics, we can sample the set of directly "illuminated" scattering surfaces (see Figure 4.12 (left)). Computing the source view using perspective projections also provides a form of importance sampling strategy by allocating more fragments to surfaces close to the source.

We can then evaluate the integral in Eq. 4.1 as a sum over all visible fragments i in this view:

$$\hat{P}(R) \approx \sum_i \hat{p}_i(R) dS_i. \tag{4.2}$$

$\hat{p}_i(R)$ is the contribution of fragment i to the integral (see [TDLD07] for details)
and $dS_i = (w/rez)^2 (nearDist/(-\mathbf{u}\cdot\mathbf{t}))^2/(\mathbf{n}\cdot\mathbf{u})$, where $\mathbf{t}$ is the viewing direction, $\mathbf{u}$ the vector from the sample to the viewpoint, $\mathbf{n}$ the normal, *nearDist* is the view plane distance, *rez* the rendering resolution and w the width of the view frustum (assuming aspect ratio is 1). All vectors are unit vectors.

Our GPU implementation renders the geometry to a floating-point offscreen render-target. For each rendered pixel, we evaluate the corresponding value of $\hat{p}_i(R)dS_i$. We store the resulting complex-valued number in two of the four color components of each pixel (see Figure 4.12 (right)).

The sum over all visible surfaces can then be efficiently computed using hierarchical integration (i.e., "mip-mapping"), classically performed in $\log(rez)/\log(k)$ render passes, where k is the reduction factor. At each pass a $k \times k$ block of values is summed-up to give a single value which will be recursively integrated in the next pass until the total value of the integral is eventually reached. In our case, we typically used a $4\times$ factor resulting in 5 passes for a 1024×1024 render-target.

To evaluate the integral over multiple frequencies, we use a deferred-shading approach [DS05] and render the necessary geometrical parameters (distances and dot products) only once. However, Eq. 4.2 must still be fully re-evaluated. For scenarios where the scattering of multiple dynamic sources must be computed at interactive rates, deferred shading can be leveraged to sample the scattering surfaces from a unique location at the expense of some visibility error. For k sources however, the process requires $k*b$ render passes of the scattering shader (Eq. 4.2) which currently limits the approach to a small number of sources or frequency components.

Using this hardware-accelerated approach, we were able to compute a full-bandwidth (0-22KHz) transfer function for the pyramid and duomo models in 2.45 Hz increments (8192 frequencies) in 92

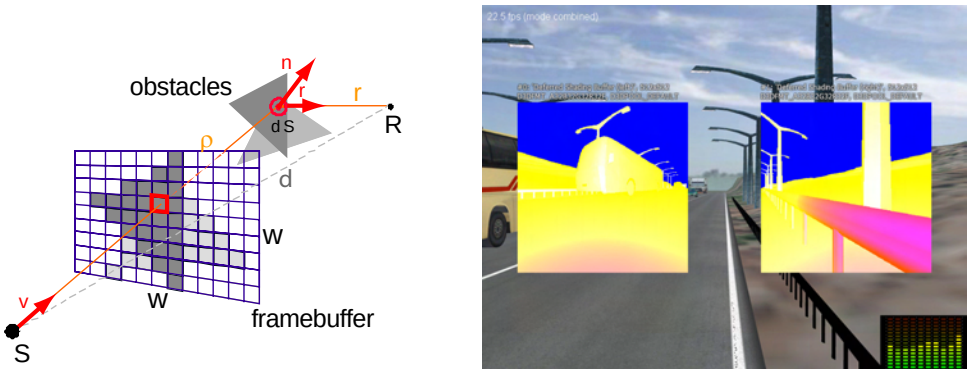


Figure 4.12: Left: Surfaces are sampled using hardware rendering from the point of view of the sound source. We evaluate the scattering terms at each pixel before global integration through mip-mapping. In this figure, S and R denote the source resp. receiver. Right: Visualisation of the scattering terms on all surfaces visible from a sound source (here, the engine of a car).

sec. on a Pentium 4 3.4GHz workstation with a GeForce 8800 GTX graphics processor. Hardware acceleration also allows for rendering direct scattering effects at interactive rates for 10 to 20 frequency subbands. Please, see [TDLD07] for details on our interactive rendering pipeline.

4.3.3 GPU-Accelerated 3D Audio Rendering

Besides geometrical calculations, GPUs can perform more general parallel operations on floating point data. In this section, we evaluate their use in the context of the signal processing required for binaural 3D audio rendering. To use GPUs for audio processing, PCM audio data must first be loaded in texture memory in a way similar to images traditionnaly used to texture 3D geometric objects.

Storing Audio Data on the GPU

The GPU memory model is targetted to 3D graphics and is different from the CPU memory model. Taking it into consideration leads to better performance. The GPU memory was designed to work with images. It is thus highly optimized for the Red-Green-Blue plus Alpha (RGBA) data type. This structure is usually packed in floating point data. To fit this model, we decompose our signals to four frequency bands, in a perceptual scale and pack them in the RGBA structure (see Figure 4.13). As a result, the four frequency bands are stored in an interleaved manner. Each pixel of the texture represents a sample of the signal.

Audio Processing

We consider a combination of two simple operations commonly used for 3D audio rendering: variable delay-line and filtering [Beg94, FJT02]. The signal of each sound source is first delayed by the propagation time of the sound wave. This involves resampling the signal at non-integer index values thus automatically accounting for Doppler shifting. The signal is then filtered to simulate the effects of source and listener directivity functions, occlusions and propagation through the medium. We resample the signals using linear interpolation between the two closest samples. On the GPU this is achieved through texture resampling. Filtering is implemented using a simple 4-band equalizer. Assuming that

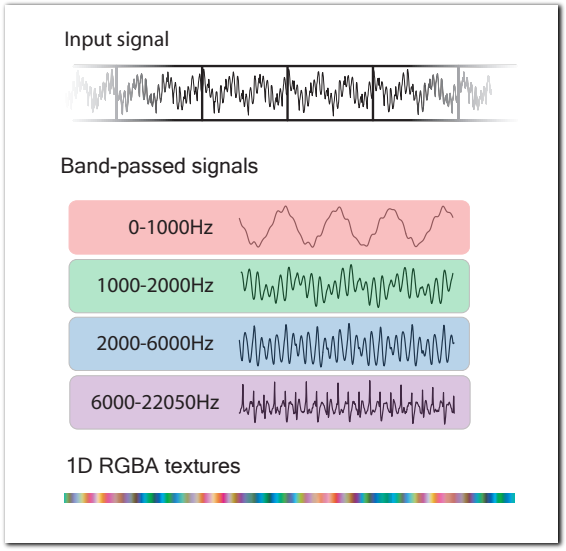


Figure 4.13: Audio data structure. (a) The incoming signal is sliced into frames. (b) The signal is decomposed into four frequency subbands. (c) the four subbands are stored in 1D RGBA textures.

input signals are band-pass filtered in a pre-processing step, the equalization is efficiently implemented as a 4-component dot product which is issued as a single GPU instruction.

Binaural stereo rendering requires applying this pipeline twice, using a direction-dependent delay and equalization for each ear, derived from head-related transfer functions (HRTFs) [Beg94]. The HRTF data is represented as an azimuth-elevation texture array (see Figure 4.15) where the RGBA component holds a gain value for the corresponding frequency band. Similar audio processing can be used to generate dynamic sub-mixes of multiple sound signals prior to spatial audio rendering (e.g., in the context of source clustering approaches [TGD04]).

Results

We compared an optimized SSE (Intel's Streaming SIMD Extensions) assembly code running on a *Pentium 4 3GHz* processor and an equivalent *Cg/OpenGL* implementation running on a *Nvidia GeForce FX 5950 Ultra* graphics board on AGP 8X and a *Nvidia Quadro FX4500* on PCI express. Audio was processed at 44.1 KHz using 1024-sample long frames. All processing was 32-bit floating point.

The GPU implementation, on a Nvidia Quadro FX4500, can perform binaural processing of up to 2250 sound sources in real time while the SSE version renders 700 sound sources in one time-frame (≈22.5 ms). However, resampling floating-point textures requires two texture fetches and a linear interpolation in the fragment shader. If floating-point texture resampling was available in hardware, GPU performance would increase. We have simulated this functionality on our GPU using a single texture-fetch and achieved real-time performance for up to 3100 sources. With the up-coming G80 processor, floating-point textures resampling is supported but not available at the time of this study.

For mono processing, the Quadro FX treats up to 6150 (1 texture fetch)/ 4580 (2 fetches and linear interp.) sources, while the CPU handles 1400 in the same amount of time. On average the GPU implementation using the Quadro FX4500 was about three times faster than the SSE implementation and

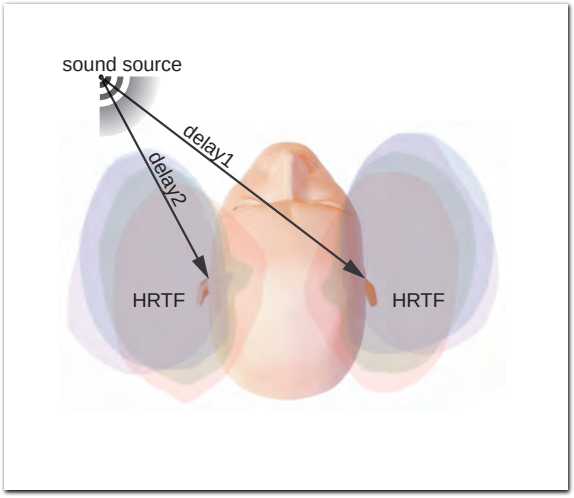


Figure 4.14: Audio processing involved in the GPU simulation. Each sound source is delayed by the propagation time and filtered to account for the distance attenuation and head-related transfer functions (HRTFs).

it would become 50% faster if floating-point texture resampling was supported in hardware. The latest graphics architectures would significantly improve GPU performance due to their increased number of pipelines and their faster RAMDAC.

The massive pixel throughput of the GPU can also be used to improve audio rendering quality without reducing frame-size by recomputing rendering parameters (source-to-listener distance, equalization gains, etc.)on a per-sample rather than per-frame basis. This can be seen as an audio equivalent of per-pixel vs. per-vertex lighting in graphics. By storing directivity functions in cube-maps and recomputing propagation delays and distances for each sample, our GPU implementation can still render up to 180 sources in the same time-frame. However, more complex texture-addressing calculations are needed in the fragment program due to limited texture size. By replacing such complex texture addressing with a single texture-fetch, we also estimated that direct support for large 1D textures would increase performance by at least a factor of 2. Novel G80 processors now support this functionality.

Running audio effects on the GPU frees-up CPU time for other tasks and can even be combined with graphics rendering with little impact on display performances for moderately graphics-demanding applications. Example movie files including GPU-generated audio and graphics are available[1]. Both audio and graphics were generated in real-time with the GPU.

4.4 Discussion

The vertex and fragment processor of the graphics hardware can be fully programmed with assembly-like languages but is not really suitable for programming complex shaders. Similar in spirit to Renderman, a shading language used by Pixar for image rendering, high level C-like languages have been introduced to program the GPUs: “CG” from Nvidia, “OpenGL Shading Language (GLSL)” from 3DLabs

[1] http://www-sop.inria.fr/reves/projects/GPUAudio/

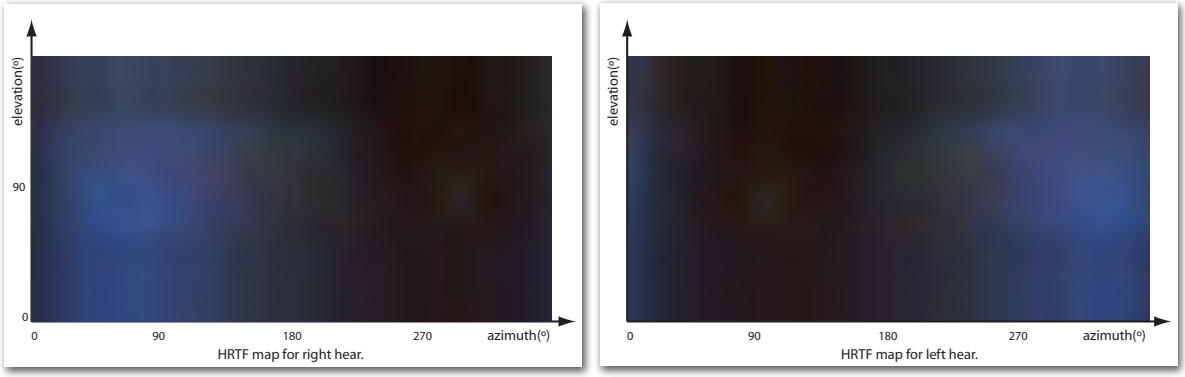


Figure 4.15: Azimuth-elevation HRTF map for the left (a) and the right ear (b). The intensity color of the RGBA component correspond to the attenuation for each frequency component generated from measured FIR data from the LISTEN HRTF database.

in conjunction with OpenGL ARB and "High Level Shading Languages (HLSL)"from Microsoft. They give the most up-to-date functionality, but graphics notions are required to program them. They are also dedicated to particular platforms API (such as OpenGL or DirectX) and graphics processor vendor. Recently, with the success of the graphics hardware as a general purpose processor unit, high level languages have emerged, proposing a friendly programming approach and hiding graphics primitives and 3D manipulation instructions.

Nvidia and ATI/AMD have recently introduced their new general purpose C-like language to get optimal performance using their latest hardware and provided total abstraction of the graphics pipeline[2].

Microsoft introduced their new general purpose language for programming graphics-processor[3].

Academic research has also introduced new alternative languages which have evolved into commercial applications. They provide development environments for programming general purpose processor including the CPU, the GPU, or the Cell processor. They offer a good compromise because they are multi-platform and independent from the hardware[4].

Finally, the parallel stream architecture introduced by the GPUs tends to abstract from the graphics and evolve towards general purpose application. Due to this architecture, the GPUs provide better performance than the CPU.

The GPUs performance has increased dramatically over the last three years compared to CPUs [OLG+05, OHL+08]. While our first experiments, in 2004, suggested that GPUs can be used for 3D audio processing with similar or increased performance compared to optimized software implementations running on top-of-the-line CPUs, the latest GPUs clearly outperform CPUs by a factor of at least 3, and, thereby, are a perfect alternative for audio processing. Moreover, GPUs surpass CPUs for a number of other tasks, including Fast Fourier Transform, a tool widely used for audio processing [BFH+04] and [GLGM06]. Figure 4.19 shows a performance comparison of the 1D Fast Fourier Transform, the CPU implementation is based on the Intel Math Kernel library and the GPU implementation based on the GPUFFTW library [GLGM06].

In our first study, we had detected several shortcomings which prevent efficient use of GPUs for mainstream audio processing applications. Due to limitations in texture-access modes and texture-size,

[2]http://www.nvidia.com/object/cuda.html, http://ati.amd.com/companyinfo/researcher/documents.html

[3]http://research.microsoft.com/research/downloads/

[4]http://www.rapidmind.net, http://www.peakstreaminc.com

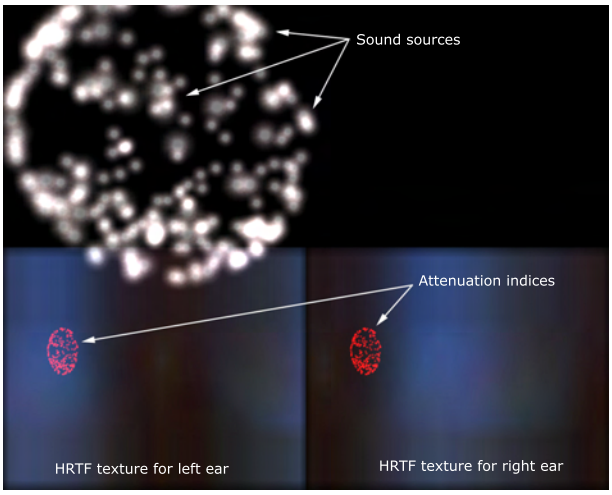


Figure 4.16: The positions of a sphere of virtual sources is mapped to an HRTF texture in the fragment program to retreive the correct attenuation coefficients.

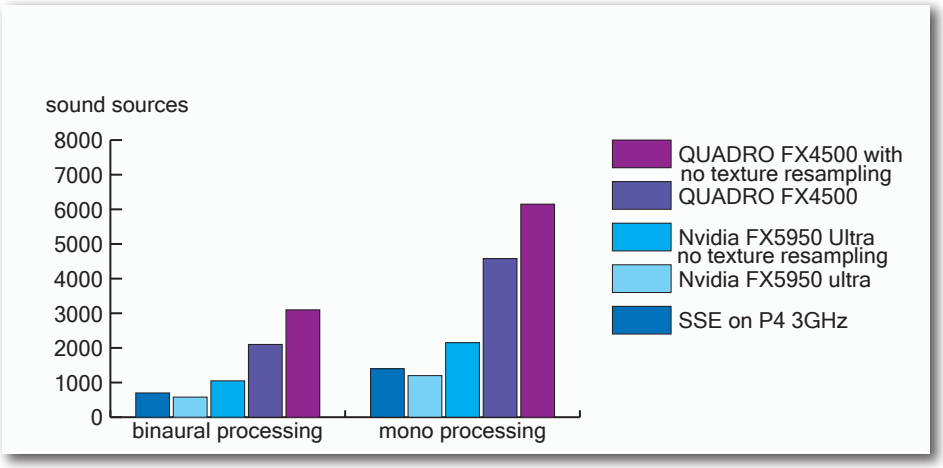


Figure 4.17: Performance tests for audio rendering on the CPU and GPU.

long 1D textures could not be easily indexed and floating-point textures resampling was not supported. The latest G80 processor now overcomes these limitations.

However, other algorithms such as infinite impulse response (recursive) filtering cannot be implemented efficiently since past values are usually unavailable when rendering a given pixel in fragment programs. As suggested in [BFH+04], including persistent registers to accumulate results across fragments would solve this problem.

On a broader scale, our results demonstrate that stream-processing architectures are appropriate for audio rendering applications and that game-audio hardware, borrowing from graphics architectures and shading languages, may benefit from including programmable “voice shaders”, enabling per-sample processing, prior to their main “effects” processor.

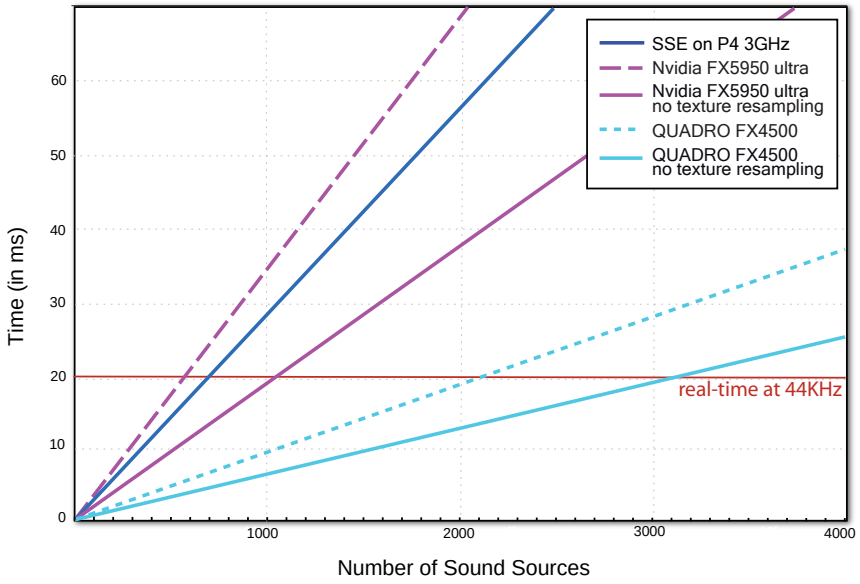


Figure 4.18: Performance for binaural audio rendering on the CPU and GPU.

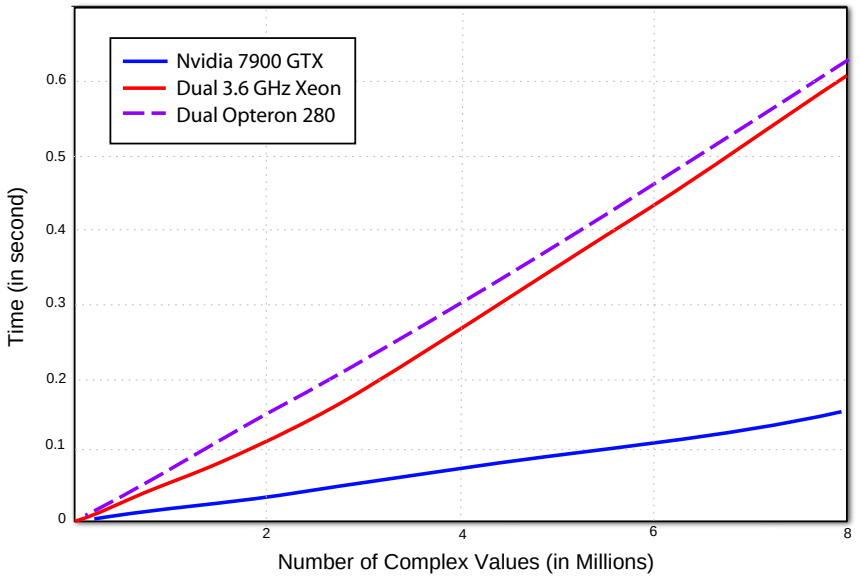


Figure 4.19: Comparison of 1D Fast Fourier Transform on CPU and GPU [GLGM06].

A beam tracing method for interactive architectural acoustics

Thomas Funkhouser[a)]
Princeton University, Princeton, New Jersey 08540

Nicolas Tsingos
INRIA, Sophia-Antipolis, France

Ingrid Carlbom
Lucent Bell Laboratories, Murray Hill, New Jersey

Gary Elko and Mohan Sondhi
Avaya Labs, Basking Ridge, New Jersey 07920

James E. West
The John Hopkins University, Baltimore, Maryland 21218

Gopal Pingali
IBM TJ Watson Research Center, Hawthorne, New York 10532

Patrick Min and Addy Ngan
Princeton University, Princeton, New Jersey 08540

(Received 9 May 2002; revised 11 April 2003; accepted 25 August 2003)

A difficult challenge in geometrical acoustic modeling is computing propagation paths from sound sources to receivers fast enough for interactive applications. This paper describes a beam tracing method that enables interactive updates of propagation paths from a stationary source to a moving receiver in large building interiors. During a precomputation phase, convex polyhedral beams traced from the location of each sound source are stored in a ''beam tree'' representing the regions of space reachable by potential sequences of transmissions, diffractions, and specular reflections at surfaces of a 3D polygonal model. Then, during an interactive phase, the precomputed beam tree(s) are used to generate propagation paths from the source(s) to any receiver location at interactive rates. The key features of this beam tracing method are (1) it scales to support large building environments, (2) it models propagation due to edge diffraction, (3) it finds all propagation paths up to a given termination criterion without exhaustive search or risk of under-sampling, and (4) it updates propagation paths at interactive rates. The method has been demonstrated to work effectively in interactive acoustic design and virtual walkthrough applications. © *2004 Acoustical Society of America.* [DOI: 10.1121/1.1641020]

PACS numbers: 43.55.Ka, 43.58.Ta [VWS] Pages: 739–756

I. INTRODUCTION

Geometric acoustic modeling tools are commonly used for design and simulation of 3D architectural environments. For example, architects use CAD tools to evaluate the acoustic properties of proposed auditorium designs, factory planners predict the sound levels at different positions on factory floors, and audio engineers optimize arrangements of loudspeakers. Acoustic modeling can also be useful for providing spatialized sound effects in interactive virtual environment systems.[1,2]

One major challenge in geometric acoustic modeling is accurate and efficient computation of propagation paths.[3] As sound travels from source to receiver via a multitude of paths containing reflections, transmissions, and diffractions (see Fig. 1), accurate simulation is extremely compute intensive. Most prior systems for geometric acoustic modeling have been based on image source methods[4,5] and/or ray tracing,[6] and therefore they do not generally scale well to support large 3D environments, and/or they fail to find all significant propagation paths containing edge diffractions. These systems generally execute in ''batch'' mode, taking several seconds or minutes to update the acoustic model for a change of the source location, receiver location, or acoustical properites of the environment,[7] and they allow visual inspection of propagation paths only for a small set of prespecified source and receiver locations.

In this paper, we describe a beam tracing method that computes early propagation paths incorporating specular reflection, transmission, and edge diffraction in large building interiors fast enough to be used for interactive applications. While different aspects of this method have appeared at computer graphics conferences,[8–10] this paper provides the first complete description of the proposed acoustic modeling system.

Briefly, our system executes as follows. During an offline precomputation, we construct a spatial subdivision in which 3D space is partitioned into convex polyhedra (cells). Later, for each sound source, we trace beams through the spatial subdivision constructing a ''beam tree'' data structure

a)Electronic mail: funk@cs.princeton.edu

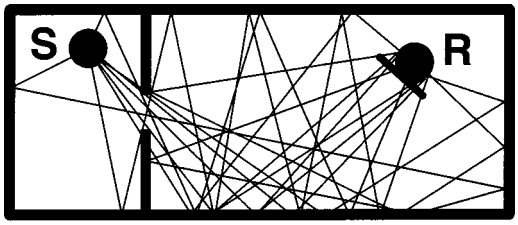


FIG. 1. Propagation paths.

encoding convex polyhedral regions of space reachable from the source by different sequences of scattering events. Then, during an interactive session, the beam trees are used to find propagation paths from the source and an arbitrary receiver location. The updates for each receiver are quick enough to be applied in an interactive acoustic design application. (For simplicity of exposition, our paper considers only propagation paths traced from sound sources to receivers. However, paths from receivers to sources could be computed just as easily—simply switch the terms ''source'' and ''receiver'' in the following text.)

The most important contribution of this paper is a method for precomputing data structures that encode potential sequences of surface scattering in a manner that enables interactive updates of propagation paths from a stationary source location to an arbitrarily moving receiver location. Our algorithms for construction and query of these data structures have the unique features that they scale well with increasing geometric complexity in densely occluded environments and that they generate propagation paths with any combination of transmission, specular reflection, and diffraction without risk of undersampling. We have incorporated these data structures and algorithms into a system that supports real-time auralization and visualization of large virtual environments.

The remainder of the paper is organized as follows. The next section reviews previous work in geometric acoustic modeling. Section III contains an overview of our system, with details of the spatial subdivision, beam tracing, path generation, and auralization methods appearing in Sec. IV. Section V contains experimental results. Applications, limitations, and topics for future work are discussed in Sec. VI. Finally, Sec. VII contains a brief conclusion.

II. PREVIOUS WORK

There have been decades of work in acoustic modeling of architectural environments, including several commercial systems for computer-aided design of concert halls (e.g., Refs. 11–13). Surveys can be found in Refs. 3 and 7.

Briefly, prior methods can be classified into two major types: (1) numerical solutions to the wave equation using finite/boundary element methods (FEM/BEM) and (2) high-frequency approximations based on geometrical propagation paths. In the latter case, image source methods, ray tracing, and beam tracing have been used to construct the sound propagation paths.

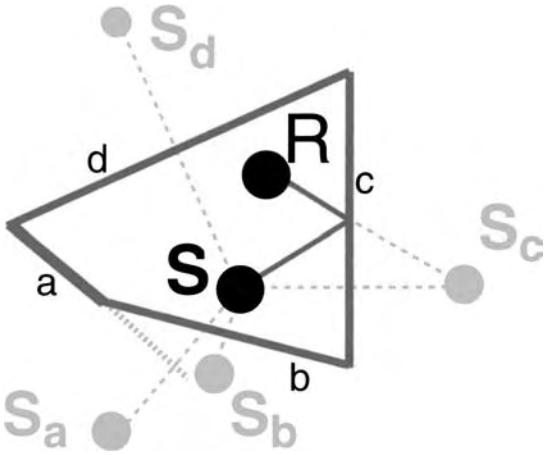


FIG. 2. Image source method.

A. Boundary element methods

Finite and boundary element methods solve the wave equation (and associated boundary conditions), subdividing space (and possibly time) into *elements*.[14–17] The wave equation is then expressed as a discrete set of linear equations for these elements. The boundary integral form of the wave equation (i.e., Green's or Helmoltz–Kirchhoff's equation) can be solved by subdividing only the boundaries of the environment and assuming the pressure (or particle velocity) is a linear combination of a finite number of basis functions on the elements. One can either impose that the wave equation is satisfied at a set of discrete points (collocation method) or ensure a global convergence criteria (Galerkin method). In the limit, finite element techniques provide an accurate solution to the wave equation. However, they are mainly used at low frequencies and for simple environments since the compute time and storage space increase dramatically with frequency.

Finite element techniques are also used to model *energy* transfer between surfaces. Such techniques have already been applied in acoustics,[18,19] as well as other fields,[20,21] and provide an efficient way of modeling diffuse global energy exchanges (i.e., where surfaces are lambertian reflectors). While they are well suited for computing energy decay characteristics in a given environment, energy exchange techniques do not allow direct reconstruction of an impulse response. Instead, they require the use of an underlying statistical model and a random phase assumption.[22] Moreover, most surfaces act primarily as specular or glossy reflectors for sound. Although extensions to nondiffuse environments have been proposed in computer graphics,[21,20] they are often time and memory consuming and not well suited to interactive applications.

B. Image source methods

Image source methods[4,5] compute specular reflection paths by considering *virtual sources* generated by mirroring the location of the audio source, S, over each polygonal surface of the environment (see Fig. 2). For each virtual source, S_i, a specular reflection path can be constructed by iterative intersection of a line segment from the source position to the

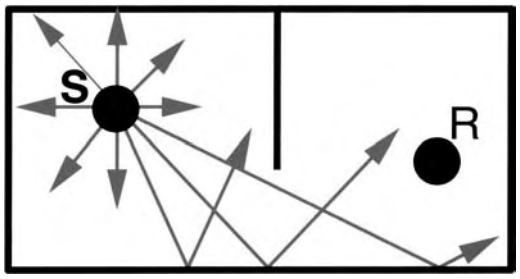


FIG. 3. Ray tracing method.

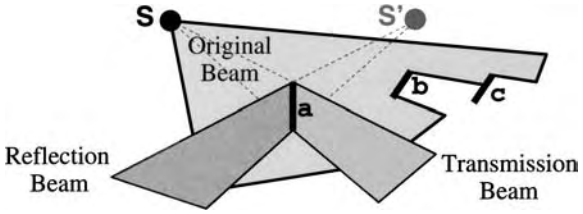


FIG. 4. Beam tracing method.

receiver position, R, with the reflecting surface planes (such a path is shown for virtual source S_c in Fig. 2). Specular reflection paths are computed up to any order by recursive generation of virtual sources.

The primary advantage of image source methods is their robustness. They guarantee that all specular paths up to a given order or reverberation time are found. However, the basic image source method models only specular reflection, and their expected computational complexity grows exponentially. In general, $O(n^r)$ virtual sources must be generated for r reflections in environments with n surface planes. Moreover, in all but the simplest environments (e.g., a box), complex validity/visibility checks must be performed for each of the $O(n^r)$ virtual sources since not all of the virtual sources represent physically realizable specular reflection paths.[5] For instance, a virtual source generated by reflection over the nonreflective side of a surface is ''invalid.''[5] Likewise, a virtual source whose reflection is blocked by another surface in the environment or intersects a point on a surface's plane which is outside the surface's boundary (e.g., S_a in Fig. 2) is ''invisible.''[5] During recursive generation of virtual sources, descendents of invalid virtual sources can be ignored. However, descendents of invisible virtual sources must still be considered, as higher-order reflections may generate visible virtual sources (consider mirroring S_a over surface d). Due to the computational demands of $O(n^r)$ visibility checks, image source methods are practical for modeling only a few specular reflections in simple environments.[23]

C. Ray tracing methods

Ray tracing methods[6] find propagation paths between a source and receiver by generating rays emanating from the source position and following them through the environment until a set of rays has been found that reach the receiver (see Fig. 3).

The primary advantage of these methods is their simplicity. They depend only on ray–surface intersection calculations, which are relatively easy to implement and have computational complexity that grows sublinearly with the number of surfaces in the model. Another advantage is generality. As each ray–surface intersection is found, paths of specular reflection, diffuse reflection, diffraction, and refraction can be sampled,[24,25] thereby modeling arbitrary types of propagation, even for models with curved surfaces. The primary disadvantages of ray tracing methods stem from their discrete sampling of rays, which may lead to undersampling errors in predicted room responses.[26] For instance, the receiver position and diffracting edges are often approximated by volumes of space (in order to enable intersections with infinitely thin rays), which can lead to false hits and paths counted multiple times.[26] Moreover, important propagation paths may be missed by all samples. In order to minimize the likelihood of large errors, ray tracing systems often generate a large number of samples, which requires a large amount of computation. Another disadvantage of ray tracing is that the results are dependent on a particular receiver position, and thus these methods are not directly applicable in interactive applications where either the source or receiver can move.

D. Beam tracing methods

Beam tracing methods[27,28] classify propagation paths from a source by recursively tracing pyramidal beams (i.e., sets of rays) through the environment (see Fig. 4). Briefly, for each beam, polygons in the environment are considered for intersection with the beam in front-to-back visibility order (i.e., such that no polygon is considered until all others that at least partially occlude it have already been considered). As intersecting polygons are detected, the original beam is clipped to remove the shadow region, a transmission beam is constructed matching the shadow region, and a reflection beam is constructed by mirroring the transmission beam over the polygon's plane. This method has been used in a variety of applications, including acoustic modeling,[27,8,29–31] illumination,[32–35,28,36] visibility determination,[37–39] and radio propagation prediction.[40,41]

The primary advantage of beam tracing is that it leverages geometric coherence, since each beam represents an infinite number of potential ray paths emanating from the source location. It does not suffer from the sampling artifacts of ray tracing,[26] nor the overlap problems of cone tracing,[42,43] since the entire space of directions leaving the source can be covered by beams exactly. The disadvantage is that the geometric operations required to trace beams through a 3D model (i.e., intersection and clipping) are relatively complex, as each beam may be reflected and/or obstructed by several surfaces.

Some systems avoid the geometric complexity of beam tracing by approximating each beam by its medial axis ray for intersection and mirror operations,[44] possibly splitting rays as they diverge with distance.[45,46] In this case, the beam representation is useful only for modeling the distribution of rays/energy with distance and for avoiding large tolerances in ray–receiver intersection calculations. If beams are not clipped or split when they intersect more than one surface, significant propagation paths can be missed, and the computed acoustical field can be grossly approximated.

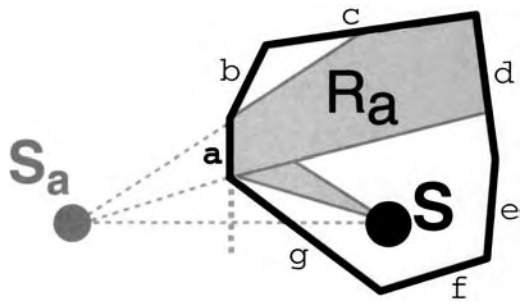


FIG. 5. Beam tracing culls invisible virtual sources.

III. OVERVIEW OF APPROACH

Our approach is based on polyhedral beam tracing. The strategy is to trace beams that decompose the space of rays into topologically distinct bundles corresponding to potential sequences of scattering events at surfaces of the 3D scene (*propagation sequences*), and then use them to guide efficient generation of propagation paths between a sound source and receiver in a later interactive phase. This approach has several advantages:

(i) **Efficient enumeration of propagation sequences**: Beam tracing provides a method for enumerating potential sequences of surface scattering events *without exhaustive search*, as in image source methods.[4,5] Since each beam describes the region of space containing all possible rays representing a particular sequence of scattering events, only surfaces intersecting the beam must be considered for further propagation. For instance, in Fig. 5, consider the virtual source S_a, which results from mirroring S over polygon a. The corresponding specular reflection beam, R_a, contains exactly the set of receiver points for which S_a is valid and visible. Similarly, R_a intersects exactly the set of polygons (c and d) for which second-order reflections are possible after specular reflection off polygon a. Other polygons (b, e, f, and g) need not be considered for higher-order propagation after specular reflection off a. As in this example, beam tracing can be used to prune the combinatorial search space of propagation sequences without resorting to sampling.

(ii) **Deterministic computation**: Beam tracing provides a method for finding potential sequences of diffracting edges and reflecting faces *without risk of errors due to under-sampling*, as in ray tracing. Since the entire 2D space of directions leaving the source can be partitioned so that every ray is in exactly one beam, beam tracing methods can guarantee finding every propagation path up to a specified termination criteria. Moreover, beams support well-defined intersections with points and edges, and thus beam tracing methods do not generate the systematic errors of ray tracing due to approximations made in intersecting infinitely thin rays with infinitely thin edges or infinitely small receiver points.[26]

(iii) **Geometric coherence**: Tracing beams can improve the efficiency of multiple ray intersection tests. In particular, once a beam has been traced along a certain sequence of surface intersections, generating a ray path from a source to a receiver following the same sequence requires only checking the ray path for intersections with the surfaces of the sequence, and the expensive computation of casting rays through a scene can be amortized over several ray paths. Beam tracing can be used not only to enumerate potential propagation sequences, but also to identify which elements of the scene can potentially be blockers for each sequence.[47,48] This information can be used to generate and check occlusion of sampled propagation paths quickly—i.e., in time proportional to the length of the sequence rather than the complexity of the scene.

(iv) **Progressive refinement**: Characteristics of the sound waves represented by beams can be used to guide priority-driven strategies.[9,49] For instance, estimates of the acoustic energy carried by different beams can be used to order beam tracing steps and to detect early termination criteria. This method is far more practical than precomputing a global visibility structure, such as the visibility skeleton,[50] which requires large amounts of compute time and storage, mostly for pairs of surfaces for which transport is insignificant. Instead, the proposed approach traces beams in priority order, finding propagation paths only as necessary for the required accuracy of the solution.

The main challenge of beam tracing is to develop methods that trace beams through 3D models robustly and efficiently and that generate propagation paths quickly. Although several data structures have been proposed to accelerate beam tracing computations, including ones based on binary space partitions,[27] cell adjacency graphs,[37,41,8,38,39] and layers of 2D triangulations,[40] no previous method models edge diffraction without sampling artifacts, and none provides interactive path updates in large 3D environments.

The key idea behind our method is to precompute and store spatial data structures that encode all possible sequences of surface and edges scattering of sound emanating from each audio source and then use these data structures to compute propagation paths to arbitrary observer viewpoints for real-time auralization during an interactive user session. Specifically, we use a precomputed polyhedral cell complex to accelerate beam tracing and a precomputed beam tree data structure to accelerate generation of propagation paths. The net result is that our method (1) scales to support large architectural environments, (2) models propagation due to edge diffraction, (3) finds all propagation paths up to a given termination criterion without exhaustive search or risk of undersampling, and (4) updates propagation paths at interactive rates. We use this system for interactive acoustic design of architectural environments.

IV. IMPLEMENTATION

Execution of our system proceeds in four distinct phases, as shown in Fig. 6. The first two phases execute

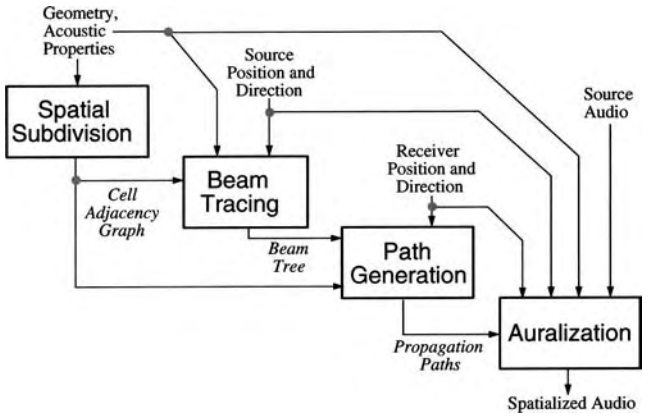


FIG. 6. System organization.

off-line, precomputing data structures for each stationary audio source, while the last two execute in real-time as a user moves the audio receiver interactively.

The result of each phase is shown in Fig. 7. First, during the *spatial subdivision phase*, we precompute spatial relationships inherent in the set of polygons describing the environment and represent them in a cell adjacency graph data structure that supports efficient traversals of space [Fig. 7(a)]. Second, during the *beam tracing phase*, we recursively follow beams of transmission, diffraction, and specular reflection through space for each audio source [Fig. 7(b)]. The output of the beam tracing phase is a beam tree data structure that explicitly encodes the region of space reachable by each sequence of reflection and transmission paths from each source point. Third, during the *path generation* phase, we compute propagation paths from each source to the receiver via lookup into the precomputed beam tree data structure as the receiver is moved under interactive user control [Fig. 7(c)]. Finally, during the *auralization phase*, we output a spatialized audio signal by convolving anechoic source signals with impulse response filters derived from the lengths, attenuations, and directions of the computed propagation paths [Fig. 7(d)]. The spatialized audio output is synchronized with real-time graphics output to provide an interactive audio/visual experience. The following subsections describe each of the four phases in detail.

A. Spatial subdivision

During the first preprocessing phase, we build a spatial subdivision representing a decomposition of 3D space and store it in a structure which we call a *winged-pair* representation. The goal of this phase is to partition space into convex polyhedral cells whose boundaries are aligned with polygons of the 3D input model and to encode cell adjacencies in a data structure enabling efficient traversals of 3D space during the later beam tracing phase.

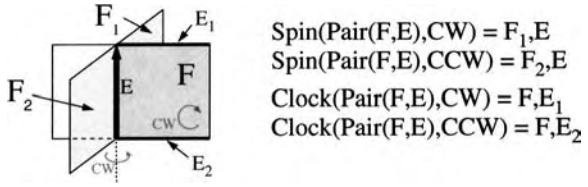


FIG. 8. Winged-pair structure.

The winged-pair data structure stores topological adjacencies in fixed-size records associated with vertices, edges, faces, cells, and face-edge pairs. Specifically, every vertex stores its 3D location and a reference to any one attached edge; every edge stores references to its two vertices and any one attached face-edge pair; every face stores references to its two cells and one attached face-edge pair; and every cell stores a reference to any one attached face. Each face-edge pair stores references to one edge *E* and one face *F* adjacent to one another, along with a fixed number of adjacency relationships useful for topological traversals. Specifically, they store references (*spin*) to the two face-edge pairs reached by spinning *F* around *E* clockwise and counter-clockwise (see Fig. 8) and to the two face-edge pairs (*clock*) reached by moving around *F* in clockwise and counter-clockwise directions from *E* (see Fig. 8). The face-edge pair also stores a bit (*direction*) indicating whether the orientation of the vertices on the edge is clockwise or counter-clockwise with respect to the face within the pair. These simple, fixed-size structures make it possible to execute efficient topological traversals of space through cell, face, edge, and vertex adjacency relationships in a manner similar to the winged-edge[51] and facet-edge structures.[52]

We build the winged-pair data structure for a 3D polygonal model using a binary space partition (BSP),[53] a recursive binary split of 3D space into convex polyhedral regions (*cells*) separated by planes. To construct the BSP, we recursively split cells by the planes of the input polygons using the method described in Ref. 54. We start with a single BSP cell containing the entire 3D space and consider polygons of the input 3D model one-by-one. For each BSP cell split by a polygon *P*, the corresponding winged-pair cell is split along the plane supporting *P*, and the faces and edges on the

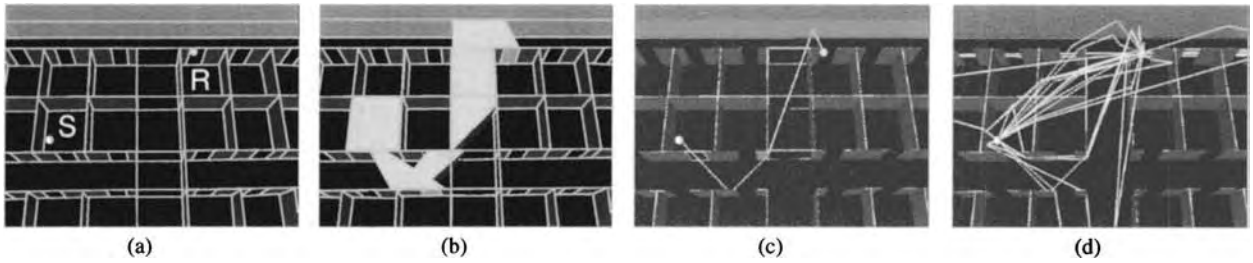


FIG. 7. Results of each phase of execution: (a) virtual environment (office cubicles) with source *S*, receiver *R*, and spatial subdivision marked in pink, (b) example reflected and diffracted beam (cyan) containing the receiver, (c) path generated for the corresponding sequence of opaque faces (green), transparent faces (purple), and edges (magenta), and (d) many paths found for different sequences from *S* to *R*.

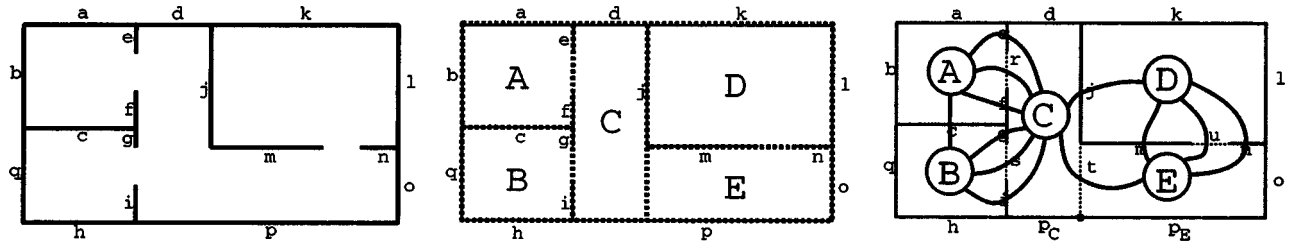


FIG. 9. Example spatial subdivision.

boundary of the cell are updated to maintain a three-manifold in which every face is flat, convex, and entirely inside or outside every input polygon. As faces are created, they are labeled according to whether they are reflectant (coincide with an input polygon) or transparent (split free space). The binary splitting process continues until no input polygon intersects the interior of any BSP cell, leading to a set of convex polyhedral cells whose faces are all convex and collectively contain all the input polygons. The resulting winged-pair is written to a file for use by later phases of our acoustic modeling process.

As an example, a simple 2D model (a) and its corresponding binary space partition (b) and cell adjacency graph (c) are shown in Fig. 9. Input ''polygons'' appear as solid line segments labeled with lower-case letters $(a-q)$; transparent cell boundaries introduced by the BSP are shown as dashed line segments labeled with lower-case letters $(r-u)$; constructed cell regions are labeled with upper-case letters $(A-E)$; and the cell adjacency graph implicit in the winged-pair structure is overlaid in Fig. 9(c).

B. Beam tracing

After the spatial subdivision has been constructed, we use it to accelerate traversals of space during beam tracing. The goal of this phase is to compute polyhedral beams representing the regions of space reachable from each stationary source by different sequences of reflections, transmissions, and diffractions. The beams are queried later during an interactive phase to compute propagation paths to specific receiver locations.

Briefly, beams are traced from each stationary sound source via a best-first traversal of the cell adjacency graph starting in the cell containing the source. As the algorithm traverses a cell boundary into a new cell, a copy of the current convex pyramidal beam is ''clipped'' to include only the region of space passing through the convex polygonal boundary to model transmissions. At each reflecting cell boundary, a copy of the transmission beam is mirrored across the plane supporting the cell boundary to model specular reflections. At each diffracting edge, a new beam is spawned whose source is the edge and whose extent includes all rays predicted by the geometric theory of diffraction.[55] The traversal along any sequence terminates when either the length of the shortest path within the beam or the cumulative attenuation exceed some user-specified thresholds. The traversal may also be terminated when the total number of beams traced or the elapsed time exceed other thresholds.

Pseudocode for the beam tracing algorithm appears in Fig. 10. Throughout the execution, a priority queue stores the set of beams to be traced sorted according to a *priority function*. Initially, the priority queue contains only one beam representing the entire space inside the cell containing the source. During each step of the algorithm, the highest priority beam B traversing a cell C is removed from the priority queue, and new ''child'' beams are placed onto the priority queue according to the following criteria:

```
void TraceBeams()
begin
    // Initialization
    S = Source point;
    D = Spatial subdivision;
    B = Beam containing all of space;
    C = Current cell;
    Q = Queue of beam tree nodes;

    C = FindCell(D, S);
    N = CreateNode(NULL, B, C);
    Q = InitQueue();

    PushQueue(Q, N);
    while (N = PopQueue(Q)) do
        // Consider each polygon on cell boundary
        foreach polygon P on boundary of N.C do
            // Check if polygon intersects beam
            if (Intersects(P, N.B)) then
                // Compute intersection beam
                Bt = Intersection(B, Beam(S, P));

                // Iterate along transmission paths
                if (Transmissive(P)) then
                    Ct = NeighborCell(D, C, P);
                    PushQueue(Q, CreateNode(N, Bt, Ct));
                endif

                // Iterate along reflection paths
                if (Reflective(P)) then
                    Br = Mirror(Bt, P);
                    PushQueue(Q, CreateNode(N, Br, C));
                endif

                // Iterate along diffraction paths
                foreach edge E on boundary of P do
                    // Check if edge intersects beam
                    if (Intersects(E, N.B)) then
                        Bd = CreateBeam(E);
                        PushQueue(Q, CreateNode(N, Bd, C));
                    endif
                endfor
            endif
        endfor
    endwhile
end
```

FIG. 10. Pseudocode for the beam tracing algorithm.

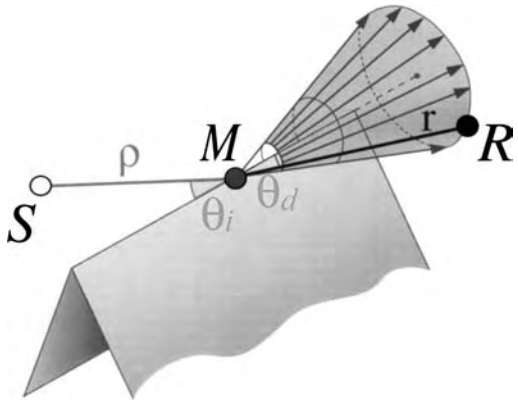


FIG. 11. Rays diffracted by a 3D edge according to the uniform theory of diffraction. The angle θ_d of the cone is equal to the angle θ_i between the incident ray and the edge.

(i) **Transmission beams**: For each nonopaque face F on the boundary of cell C and intersected by the beam B, a pyramidal *transmission beam* B_t is constructed whose apex is the source of B and whose sides each contain an edge of $F \cap B$. This new beam B_t represents sound rays traveling along B that are transmitted through F into the cell C_t which is adjacent to C across F.

(ii) **Specular reflection beams**: For each reflective face F on the boundary of cell C and intersected by the beam B, a polyhedral *specular reflection beam* B_r is constructed whose apex is the virtual source of B, created by mirroring the source of B over the plane containing F, and whose sides each contain an edge of $F \cap B$. This new beam B_r represents sound rays traveling along B that reflect specularly off of F and back into cell C.

(iii) **Diffraction beams**: For each edge E shared by two scattering faces F_1 and F_2 on the boundary of cell C and intersected by beam B, a *diffraction beam* is formed whose source is the line segment describing E and whose polyhedral extent contains the cone of potential diffraction paths bounded by the solid wedge of opaque surfaces sharing E, as shown in Fig. 11. This conservatively approximate beam contains all potential paths of sound initially traveling along B and then diffracted by edge E. For efficiency, the user may specify that diffraction beams should be traced only into shadow regions, in which case an extra half-space representing the shadow boundary is added to the beam.

Figure 12 contains an illustration of the beam tracing algorithm execution for the simple 2D example model shown in Fig. 9. The best-first traversal starts in the cell (labeled "D") containing the source point (labeled "S") with a beam containing the entire cell (D). Beams are created and traced for each of the six boundary polygons of cell "D" (j, k, l, m, n, and u). For example, transmission through the cell boundary labeled "u" results in a beam (labeled T_u) that is trimmed as it enters cell "E." T_u intersects only the polygon labeled "o," which spawns a reflection beam (labeled $T_u R_o$).

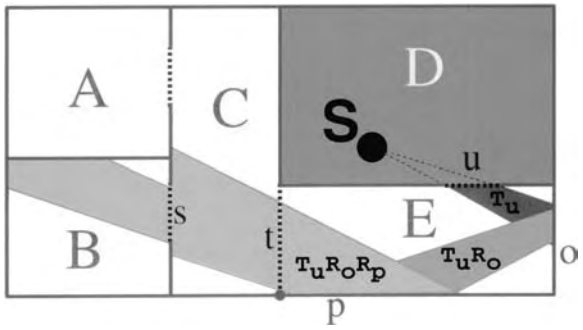


FIG. 12. Beam tracing through cell adjacency graph (this figure shows only one beam, while many such beams are traced along different propagation sequences).

That beam intersects only the polygon labeled "p," which spawns a reflection beam (labeled $T_u R_o R_p$), and so on.

Figure 13 shows an example in 3D with one sequence of beams traced up to one reflection from a source (on left) through the spatial subdivision (thin lines are cell boundaries) for a simple set of input polygons.

If the source is not a point, but instead distributed in a region of space (e.g., for diffracting edges), the exact region of space reachable by rays transmitted or reflected by a sequence of convex polygons can become quite complex, bounded by quadric surfaces corresponding to triple-edge (EEE) events.[56] Rather than representing these complex regions exactly, we conservatively overestimate the potential space of paths from each region of space edge with a convex polyhedron bounded by a fixed number of planes (usually six, as in Ref. 39). We correct for this approximation later during path generation by checking each propagation path to determine if it lies in the overestimating part of the polyhedron, in which case it is discarded. Since propagation patterns can be approximated conservatively and tightly with simple convex polyhedra, and since checking propagation paths is quick, the whole process is much more robust and faster than computing the exact propagation pattern directly. Using the adjacency information in the winged-pair structure, each new beam is constructed in constant time.

The results of the beam tracing algorithm are stored in a *beam tree* data structure[28] to be used later during path generation for rapid determination of propagation paths from the

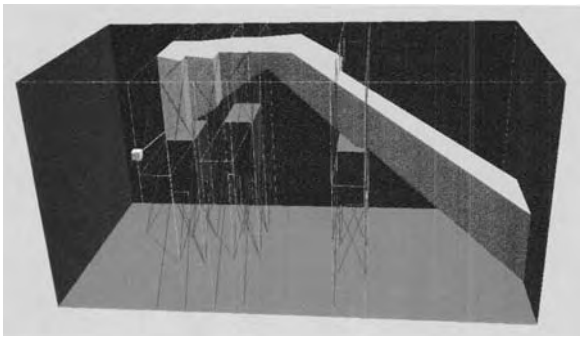


FIG. 13. A beam clipped and reflected at cell boundaries (this figure shows only one beam, while many such beams are traced along different propagation sequences).

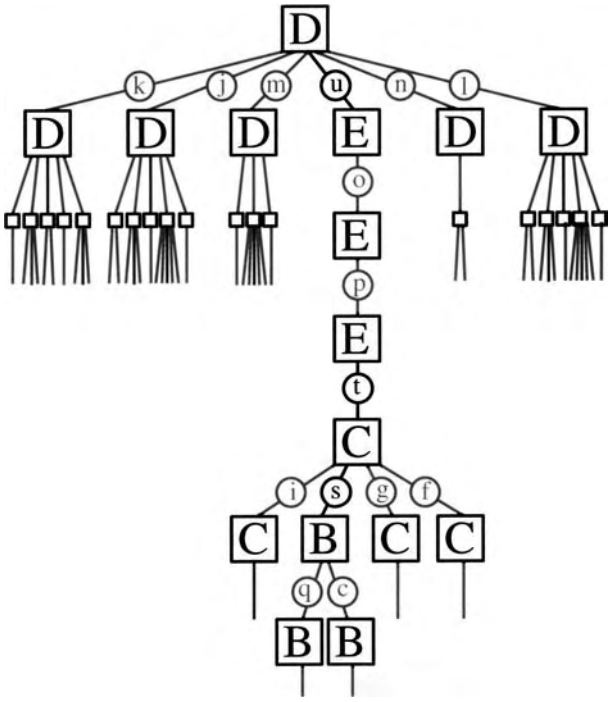


FIG. 14. Beam tree.

source point. The beam tree contains a node for each beam considered during the beam tracing algorithm. Specifically, each node stores (1) a reference to the cell being traversed, (2) a reference to the edge/face most recently traversed (if there is one), and (3) the convex polyhedral beam representing the region of space potentially reachable by the traversed sequence of transmissions, reflections, and diffractions. To further accelerate evaluation of propagation paths during a later interactive phase, each node of the beam tree also stores the cumulative attenuation due to reflective and transmissive absorption, and each cell of the spatial subdivision stores a list of ''back-pointers'' to its beam tree nodes. Figure 14 shows a partial beam tree corresponding to the traversal shown in Fig. 12.

C. Path generation

In the third phase, as a user moves the receiver interactively through the environment, we use the precomputed beam trees to identify propagation sequences of transmissions, reflections, and diffractions potentially reaching the receiver location.

Since every beam contains all points potentially reachable by rays traveling along a particular propagation sequence, we can quickly enumerate the potential propagation sequences by finding all the beams containing the receiver location. Specifically, we first find the cell containing the receiver by a logarithmic-time search of the BSP. Then, we check each beam tree node, T, associated with that cell to see whether the beam stored with T contains the receiver. If it does, a potential propagation sequence from the source point to the receiver point has been found, and the ancestors of T in the beam tree explicitly encode the set of reflections, diffractions, and transmissions through the boundaries of the

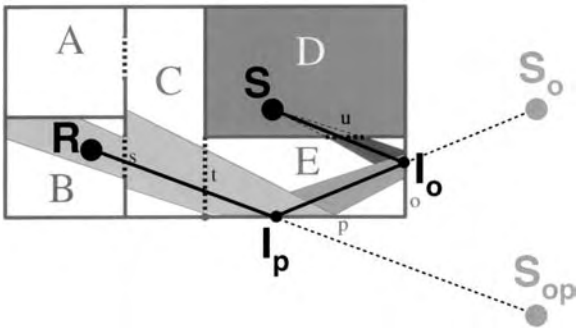


FIG. 15. Propagation path to receiver point (''R'') for the example in Figs. 12 and 14 computed via lookup in beam tree for source point (''S'').

spatial subdivision that a ray must traverse from the source to the receiver along this sequence (more generally, to any point inside the beam stored with T).

For each such propagation sequence, we construct explicit propagation path(s) from the source to the receiver. In our current system, we compute a single propagation path for each sequence as the one that is the shortest among all possible piecewise-linear paths from the source to the receiver (this path is used directly for modeling transmission, specular reflection, and diffraction according to the geometrical theory of diffraction). In order to construct this shortest path, we must find the points of intersection of the path with every face and edge in the sequence.

For sequences containing only transmissions and specular reflections (i.e., no edge diffractions), the shortest propagation path is generated analytically by iterative intersection with each reflecting surface. For instance, to find a path between a specific pair of points, S and R, along a sequence of specularly reflecting polygons P_i for $i=1,...,n$, we first traverse the polygon sequence in forward order to construct a stack of mirror images of S, where S_i corresponds to the image resulting from mirroring S over the first i of the n reflecting polygons in the sequence. Then, we construct the propagation path by traversing the polygon sequence in backward order, computing the ith vertex, V_i, of the path as the intersection of the line between V_{i-1} and S_{n-i+1} with the surface of polygon P_{n-i+1}, where V_0 is the receiver point. If every vertex V_i of the path lies within the boundary of the corresponding polygon P_i, we have found a *valid* reflection path from S to R along P. Otherwise, the path is in an overestimating part of the beam, and it can be ignored. Figure 15 shows the valid specular reflection path from the source (labeled ''S'') to a receiver (labeled ''R'') for the example shown in Fig. 12.

For sequences also containing diffracting edges, construction of the shortest propagation path is more difficult since it requires determining the locations of ''diffraction points,'' $D_i\,(i=1,...,n)$, for the n diffracting edges. These diffraction points generally lie in the interior of the diffracting edges (see Fig. 16), and the path through them locally satisfies a simple ''unfolding property:'' the angle (θ_i) at which the path enters each diffracting edge must be the same as the angle (ϕ_i) at which it leaves.[57] (The unfolding property is a consequence of the generalized Fermats principle.[55]) Thus, to find the shortest path through n diffracting edges

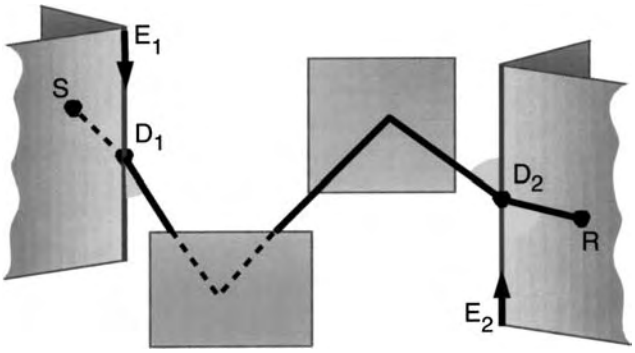


FIG. 16. A single propagation path comprising a diffraction, two specular reflections, and another diffraction. The two diffraction points (D_i) are determined by equal angle constraints at the corresponding edges (E_i).

and m transmitting and/or specularly reflecting faces, we must solve a nonlinear system of n equations expressing equal angle constraints at all diffracting edges:

$$\begin{aligned}
\overrightarrow{D_1S}\cdot\vec{E_1} &= \overrightarrow{D_1D_2}\cdot(-\vec{E_1})\\
\overrightarrow{D_2D_1}\cdot\vec{E_2} &= \overrightarrow{D_2D_3}\cdot(-\vec{E_2})\\
&\vdots\\
\overrightarrow{D_nD_{n-1}}\cdot\vec{E_n} &= \overrightarrow{D_nR}\cdot(-\vec{E_n})
\end{aligned} \tag{1}$$

where S is the source position, R is the receiver position, $\vec{E_i}$ is the normalized direction vector of the ith diffracting edge, and $\overrightarrow{D_{i+1}D_i}$ is a normalized direction vector between two adjacent points in the shortest path. To incorporate specular reflections in this equation, $\vec{E_i}$ and $\overrightarrow{D_{i+1}D_i}$ are both transformed by a mirroring operator accounting for the sequence of specularly reflecting faces up to the ith diffraction.

Parametrizing the edges, $D_i = O_i + t_i\vec{E_i}$ (where O_i is a reference point on edge i), the system of equations (1) can be rewritten in terms of n unknowns (t_i) and solved within a specified tolerance using a nonlinear system solving scheme. We use a locally convergent Newton scheme,[58] with the middle of the edges as a starting guess for the diffraction points. Since the equation satisfied by any diffraction point only depends on the previous and next diffraction points in the sequence, the Jacobian matrix is tridiagonal and can easily be evaluated analytically. Thus, every Newton iteration can be performed in time $O(n)$ where n is the number of unknowns (i.e., edges). We found this method to be faster than the recursive geometrical construction proposed by Aveneau.[59]

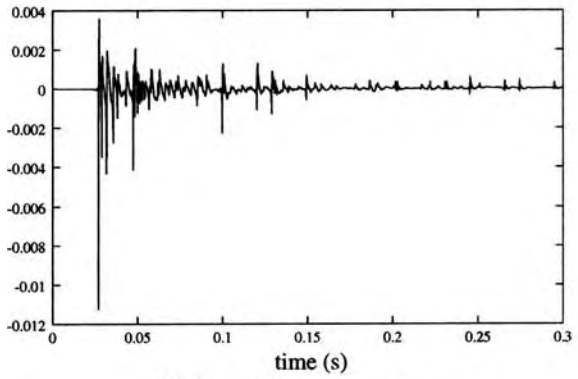
Once the intersection points of a propagation path are found, we validate whether the path intersects every surface and edge in the sequence (to compensate for the fact that the beams are conservatively approximate). If not, the path belongs to the overestimating part of the beam and is discarded. Otherwise, it contributes to an *impulse response* used for spatializing sound.

D. Auralization

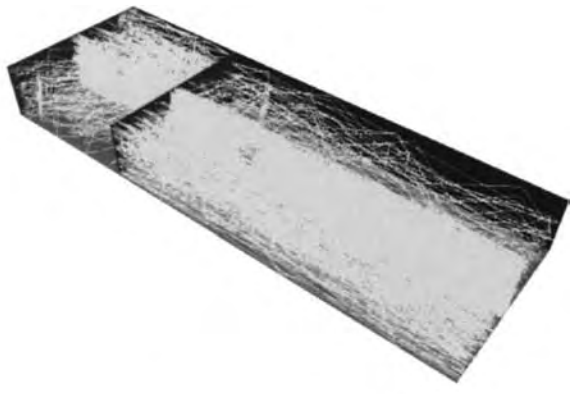
The computed early propagation paths are combined with a statistical approximation of late reverberation to model an impulse response of the virtual environment for every source/receiver pair. Dry sound signals emanating from the source are convolved with this digital filter to produce spatialized audio output.

Although this paper focuses on computation of geometric propagation paths, for the sake of completeness, we describe two auralization methods implemented in our system: (1) an off-line, high-resolution method for applications in which accuracy is favored over speed, and (2) an on-line, low-resolution approximation suitable for interactive walkthrough applications. Please refer to other papers (e.g., Refs. 7 and 60) for more details on auralization methods.

In the off-line case, we compute the early part of the impulse response in the Fourier frequency domain at the sampling rate resolution (e.g., 8000 complex values are updated for every propagation path for a one second long response at 16 kHz). As an example, Fig. 17 shows an impulse response computed for up to ten orders of specular reflections between source and receiver points in coupled-rooms. Our implementation includes frequency-dependent source and head filtering effects (obtained through measurements) and material filtering effects (derived from either measure-



(a) Impulse response



(b) Propagation paths

FIG. 17. Impulse response (left) computed for up to ten orders of specular reflections (right) between a point source (B&K "artificial mouth") and point receiver (omnidirectional) in a coupled-rooms environment (two rooms connected by an open door). There are 353 paths. The small room is $7\times8\times3$ m^3, while the large room is $17\times8\times3$ m^3.

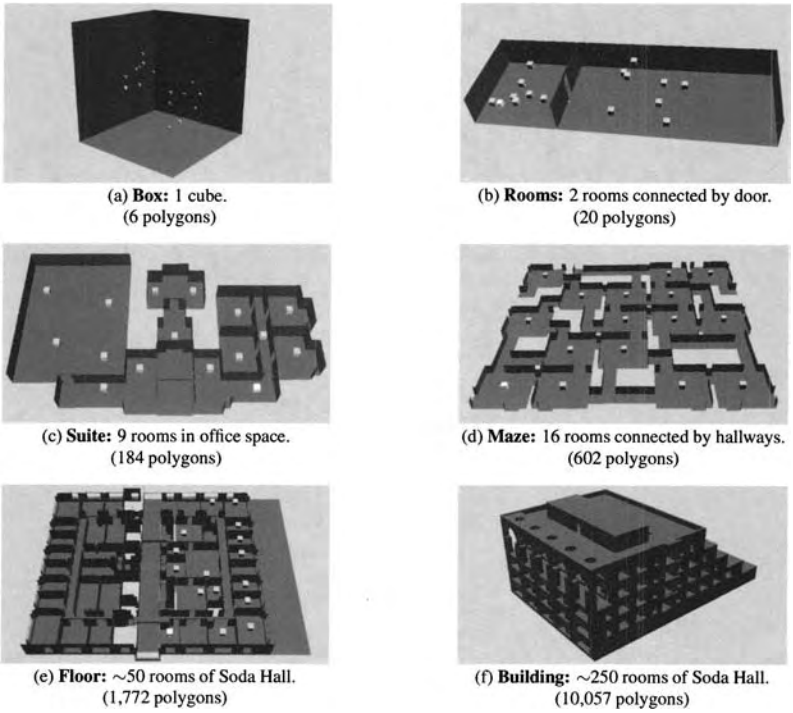


FIG. 18. Test models (source locations are gray dots).

ments or analytical models). We derive analytical models for the frequency-dependent impedance from the Delany–Bazley formula[61] and for the pressure reflection coefficient from the well-known plane wave formula of Pierce[62] (p. 33). Other wave formulas, such as the exact expression for the reflection of a spherical wave off an infinite impedant surface, derived by Thomasson,[63] could be used for improved accuracy in the near field from the surfaces.

We compute frequency-dependent diffraction coefficients using the uniform theory of diffraction[55,64,65] applied along the shortest paths constructed by our algorithm. If more accuracy is required, the information given in computed propagation sequences (exact intersected portions of surfaces and edges) can be used to derive filters, for example based on more recent results exploiting the Biot–Tolstoy–Medwin approach.[66–69] In this case, the shortest path computed by our algorithms can still be used to determine efficiently a unique incident direction on the listener's head for binaural processing (as suggested in Ref. 69).

In the on-line case, our system auralizes sound in real-time as the receiver position moves under interactive user control. A separate, concurrently executing process is spawned to perform convolution in software. To provide plausible spatialization with limited processing resources, we use a small number of frequency bands to reequalize and delay the source signal for every path, computing its contribution to a stereo impulse response in the time domain.[60] The delay associated with each path is given by L/C, where L is the length of the corresponding propagation path, and C is the speed of sound. The amplitude is given by A/L, where A is the product of all the attenuation coefficients for the reflecting, diffracting, and transmitting surfaces along the corresponding propagation sequence. Stereo impulse responses are generated by multiplying the amplitude of each path by the cardioid directivity function $((1+\cos(\theta))/2)$, where θ is the angle of arrival of the pulse with respect to the normal vector pointing out of the ear) corresponding to each ear. These gross approximations enable our auralization to give real-time feedback with purely software convolution. Other methods utilizing DSP hardware (e.g., binaural presentation) could easily be incorporated into our system in the future.

V. RESULTS

The 3D data structures and algorithms described in the preceding sections have been implemented in C++ and run on Silicon Graphics and PC/Windows computers.

To test whether the algorithms support large 3D environments and update propagation paths at interactive rates, we performed a series of experiments in which propagation paths were computed in a variety of architectural models (shown in Fig. 18). The test models ranged from a simple box with 6 polygons to a complex building with over 10 000 polygons. The experiments were run on a Silicon Graphics Octane workstation with 640 MB of memory and used one 195 MHz R10000 processor.

The focus of the experiments is to compare the computational efficiency of our method with image source methods, the approach most commonly used for interactive acoustic modeling applications. Accordingly, for the sake of direct comparison, we limited our beam tracing system to consider only specular reflections. In this case, our beam tracing method produces exactly the same set of propagation paths as classical image source methods. However, as we shall see, our beam tracing method has the ability to scale to large environments and to generate propagation paths at interactive rates.

In each experiment, we measured the time and storage required for spatial subdivision, beam tracing, sequence con-

TABLE I. Spatial subdivision statistics.

Model name	Input polys	Cell regions	Cell boundaries	Time (s)	Storage (MB)
Box	6	7	18	0.0	0.004
Rooms	20	12	43	0.1	0.029
Suite	184	98	581	3.0	0.352
Maze	602	172	1187	4.9	0.803
Floor	1772	814	5533	22.7	3.310
Bldg	10 057	4512	31 681	186.3	18.694

struction, and path generation. Results are reported in the following subsections.

A. Spatial subdivision results

We first constructed the spatial subdivision data structure (cell adjacency graph) for each test model. Statistics from this phase of the experiment are shown in Table I. Column 2 lists the number of input polygons in each model, while columns 3 and 4 contain the number of cell regions and boundary polygons, respectively, generated by the spatial subdivision algorithm. Column 5 contains the wall-clock execution time (in seconds) for the algorithm, while column 6 shows the storage requirements (in MBs) for the resulting spatial subdivision.

Empirically, we find that the number of cell regions and boundary polygons grows linearly with the number of input polygons for typical architectural models (see Fig. 19), rather than quadratically as is possible for worst case geometric arrangements. The reason for linear growth is illustrated in the two images inlaid in Fig. 19, which compare spatial subdivisions for the Maze test model (on the left) and a 2×2 grid of Maze test models (on the right). The 2×2 grid of Mazes has exactly four times as many polygons and approximately four times as many cells. The storage requirements of the spatial subdivision data structure also grow linearly as they are dominated by the vertices of boundary polygons.

The time required to construct the spatial subdivisions grows super-linearly, dominated by the code that selects and orders splitting planes during BSP construction (see Ref. 54). However, it is important to note that the spatial subdivision phase need be executed only once off-line for each geometric model, as its results are stored in a file, allowing rapid reconstruction in subsequent beam tracing executions.

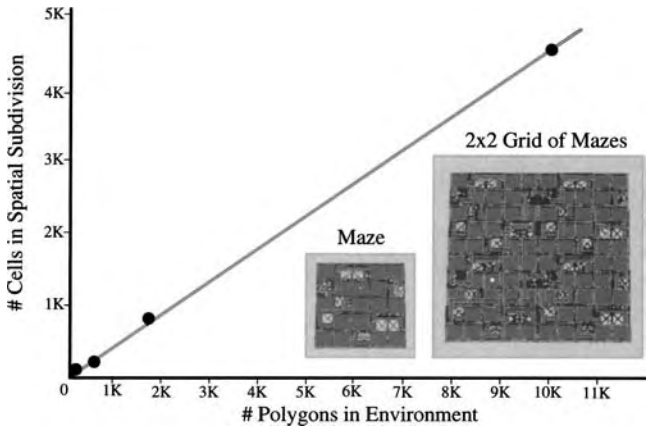


FIG. 19. Plot of subdivision size versus polygonal complexity.

TABLE II. Beam tracing and path generation statistics.

Model name	No. of polys	No. of Rfl	Beam tracing		Path generation	
			No. of beams	Time (ms)	No. of paths	Time (ms)
Box	6	0	1	0	1.0	0.0
		1	7	1	7.0	0.1
		2	37	3	25.0	0.3
		4	473	42	129.0	6.0
		8	10 036	825	833.0	228.2
Rooms	20	0	3	0	1.0	0.0
		1	31	3	7.0	0.1
		2	177	16	25.1	0.3
		4	1939	178	127.9	5.2
		8	33 877	3024	794.4	180.3
Suite	184	0	7	1	1.0	0.0
		1	90	9	6.8	0.1
		2	576	59	25.3	0.4
		4	7217	722	120.2	6.5
		8	132 920	13 070	672.5	188.9
Maze	602	0	11	1	0.4	0.0
		1	167	16	2.3	0.0
		2	1162	107	8.6	0.1
		4	13 874	1272	36.2	2.0
		8	236 891	21 519	183.1	46.7
Floor	1772	0	23	4	1.0	0.0
		1	289	39	6.1	0.1
		2	1713	213	21.5	0.4
		4	18 239	2097	93.7	5.3
		8	294 635	32 061	467.0	124.5
Bldg	10 057	0	28	5	1.0	0.0
		1	347	49	6.3	0.1
		2	2135	293	22.7	0.4
		4	23 264	2830	101.8	6.8
		8	411 640	48 650	529.8	169.5

B. Beam tracing results

We tested our beam tracing algorithm with 16 source locations in each test model. The source locations were chosen to represent typical audio source positions (e.g., in offices, in common areas, etc.)—they are shown as gray dots in Fig. 18 (we use the same source locations in Building model as in the Floor model). For each source location, we traced beams (i.e., constructed a beam tree) five times, each time with a different limit on the maximum order of specular reflections (e.g., up to 0, 1, 2, 4, or 8 orders). Other termination criteria based on attenuation or path length were disabled, and transmission was ignored, in order to isolate the impact of input model size and maximum order of specular reflections on computational complexity.

Table II contains statistics from the beam tracing experiment—each row represents a test with a particular 3D model and maximum order of reflections, averaged over all 16 source locations. Columns 2 and 3 show the number of polygons describing each test model and the maximum order of specular reflections allowed in each test, respectively. Column 4 contains the average number of beams traced by our

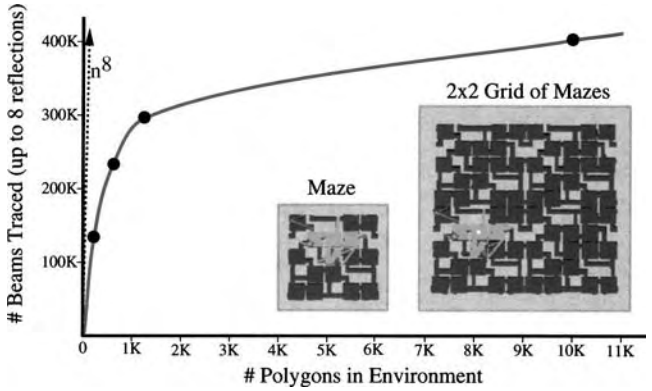


FIG. 20. Plot of beam tree size versus polygonal complexity.

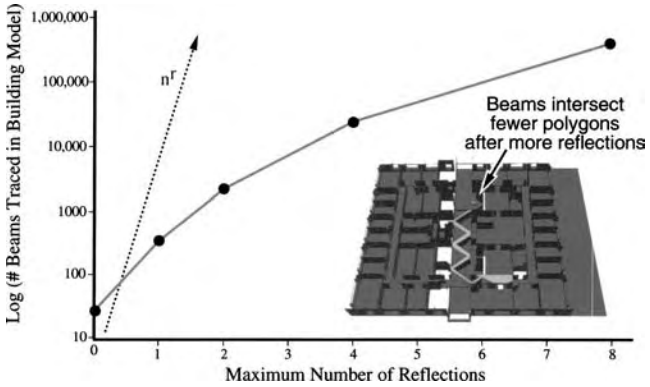


FIG. 21. Plot of beam tree size with increasing reflection orders.

algorithm (i.e., the average number of nodes in the resulting beam trees), and column 5 shows the average wall-clock time (in milliseconds) for the beam tracing algorithm to execute.

1. Scale with increasing polygonal complexity

We readily see from the results in column 4 that the number of beams traced by our algorithm (i.e., the number of nodes in the beam tree) does *not* grow at an exponential rate with the number of polygons (n) in these environments (as it does using the image source method). Each beam traced by our algorithm preclassifies the regions of space according to whether the corresponding virtual source (i.e., the apex of the beam) is visible to a receiver. Rather than generating a virtual source (beam) for every front-facing surface at each step of the recursion as in the image source method, we directly find only the potentially visible virtual sources via beam-polygon intersection and cell adjacency graph traversal. We use the current beam and the current cell of the spatial subdivision to find the small set of polygon reflections that admit visible higher-order virtual sources.

The benefit of this approach is particularly important for large environments in which the boundary of each convex cell is simple, and yet the entire environment is very complex. As an example, consider computation of up to eighth order specular reflections in the Building test model (the last row of Table II). The image source method must consider approximately 1 851 082 741 virtual sources ($\Sigma_{r=0}^{8}(10\,057/2)^r$), assuming half of the 10 057 polygons are front-facing to each virtual source. Our beam tracing method considers only 411 640 virtual sources, a difference of four orders of magnitude. In most cases, it would be impractical to build and store the recursion tree without such effective pruning.

In ''densely occluded'' environments, in which all but a little part of the environment is occluded from any source point (e.g., most buildings and cities), the number of beams traced by our algorithm even grows sublinearly with the total number of polygons in the environment (see Fig. 20). In these environments, the number of sides to each polyhedral cell is nearly constant, and a nearly constant number of cells are reached by each beam, leading to near-constant expected-case complexity of our beam tracing algorithm with increasing global environment complexity.

This result is most readily understood by comparing the number of beams traced for up to eighth order reflections in the Floor and Building models (i.e., the rightmost two data points in Fig. 20). The Floor model represents the fifth floor of Soda Hall at UC Berkeley. It is a subset of the Building model, which represents five floors of the same building (floors 3–7 and the roof). Although the Building model (10 057 polygons) has more than five times the complexity of the Floor model (1772 polygons), the average number of beams traced from the same source locations by our algorithm is only 1.2–1.4 times larger for the Building model (e.g., 411 640/294 635=1.4). This is because the complexity of the spatial subdivision on the fifth floor of the building is similar in both cases, and most other parts of the building are not reached by any beam. Similarly, we expect that the beam tracing algorithm would have nearly the same complexity if the entire building were 1000 floors high, or if it were in a city of 1000 buildings. This result is shown visually in Fig. 20: the number of beams (green) traced in the Maze test model (left) does not increase significantly if the model is increased to be a 2×2 grid of Maze models (right). *The beam tracing algorithm is impacted only by local complexity, and not by global complexity.*

2. Scale with increasing reflections

We see that the number of beams traced by our algorithm grows exponentially as we increase the maximum order of reflections (r), but far slower than $O(n^r)$ as in the image source method. Figure 21 shows a logscale plot of the average number of beams traced in the Building model with increasing orders of specular reflections. The beam tree growth is less than $O(n^r)$ because each beam narrows as it is clipped by the cell boundaries it has traversed, and thus it tends to intersect fewer cell boundaries (see the example beam inlaid in Fig. 21). In the limit, each beam becomes so narrow that it intersects only one or two cell boundaries, on average, leading to a beam tree with a small branching factor (rather than a branching factor of $O(n)$, as in the image source method).

As an example, consider Table III which shows the average branching factor for nodes at each depth of the beam tree constructed for up to eighth order specular reflections in the Building model from one source location. The average

TABLE III. Example beam tree branching statistics. The number of interior nodes (have children) and leaf nodes (have no children) at each depth are listed, along with the average branching factor for interior nodes.

Tree depth	Total nodes	Interior nodes	Leaf nodes	Branching factor
0	1	1	0	16.0000
1	16	16	0	6.5000
2	104	104	0	4.2981
3	447	446	1	2.9193
4	1302	1296	6	2.3920
5	3100	3092	8	2.0715
6–10	84 788	72 469	12 319	1.2920
11–15	154 790	114 664	40 126	1.2685
>15	96 434	61 079	35 355	1.1789

branching factor (column 5) generally decreases with tree depth and is generally bounded by a small constant in lower levels of the tree.

C. Path generation results

In order to verify that specular reflection paths are computed at interactive rates from stationary sources as the receiver moves, we conducted experiments to quantify the complexity of generating specular reflection paths to different receiver locations from precomputed beam trees. For each beam tree in the previous experiment, we logged statistics during generation of specular propagation paths to 16 different receiver locations. Receivers were chosen randomly within a two foot sphere around the source to represent a typical audio scenario in which the source and receiver are in close proximity within the same ''room.'' We believe this represents a worst-case scenario as fewer paths would likely reach more remote and more occluded receiver locations.

Columns 6 and 7 of Table II contain statistics gathered during path generation for each combination of model and termination criterion averaged over all 256 source-receiver pairs (i.e., 16 receivers for each of the 16 sources). Column 6 contains the average number of propagation paths generated, while column 7 shows the average wall-clock time (in milliseconds) for execution of the path generation algorithm. Figure 22 shows a plot of the wall-clock time required to generate up to eighth order specular reflection paths for each test model.

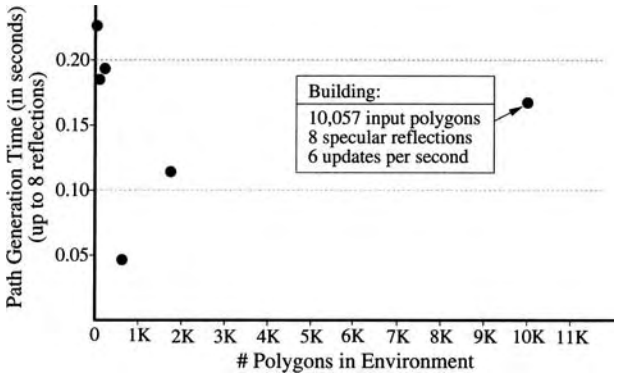


FIG. 22. Path compute time versus polygonal complexity.

We find that the number of specular reflection paths is nearly constant across all of our test models when a source and receiver are located in close proximity of one another. Also, the time required by our path generation algorithm is generally *not* dependent on the number of polygons in the environment (see Fig. 22), nor is it dependent on the total number of nodes in the precomputed beam tree. This result is due to the fact that our path generation algorithm considers only nodes of the beam tree with beams residing inside the cell containing the receiver location. Therefore, the computation time required by the algorithm is *not* dependent on the complexity of the environment outside the receiver's cell, but instead on the number of beams that traverse the receiver's cell.

Overall, we find that our algorithm supports generation of specular reflection paths between a fixed source and any (arbitrarily moving) receiver at interactive rates in complex environments. For instance, we are able to compute up to eighth order specular reflection paths in the Building environment with more than 10 000 polygons at a rate of approximately six times per second (i.e., the rightmost point in the plot of Fig. 22).

VI. DISCUSSION

In this paper, we describe beam tracing algorithms and data structures that accelerate computation of propagation paths in large architectural environments. The following subsections discuss applications of the proposed methods, limitations of our approach, and related topics for further study.

A. Applications

There are several potential applications for the methods proposed in this paper. For instance, traditional accoustical design programs (e.g., CATT Acoustics[12]) could be enhanced with real-time auralization and visualization that aid a user in understanding which surfaces cause particular acoustical effects.

Alternatively, real-time acoustic simulation can be used to enhance simulation of virtual environments in interactive walkthrough applications. Auditory cues are important in immersive applications as they can combine with visual cues to

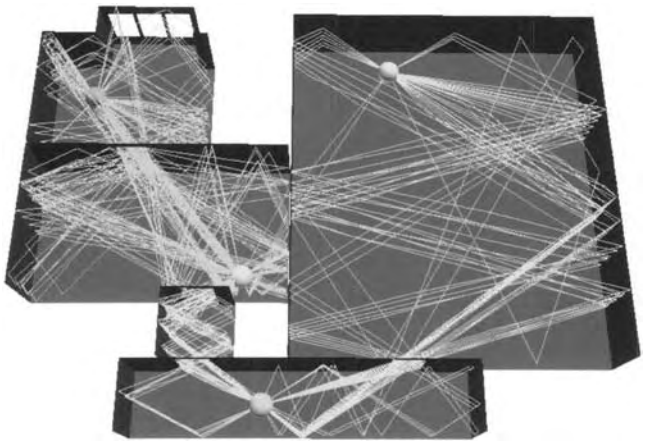


FIG. 23. Sound propagation paths (lines) between four avatars (spheres) representing users in shared virtual environment.

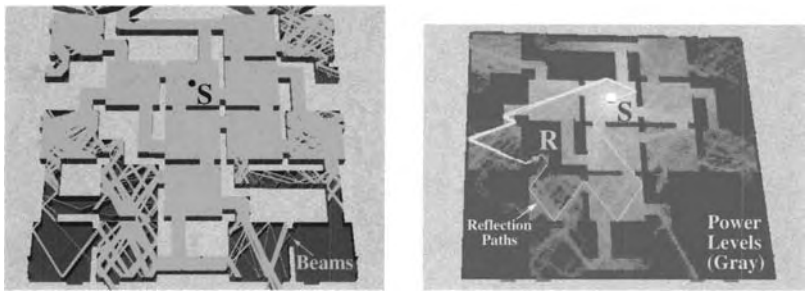


FIG. 24. Eighth-order specular reflection beams (left) and predicted power levels (right) in Maze model.

aid localization of objects, separation of simultaneous sound signals, and formation of spatial impressions of an environment.[70] For instance, binaural auditory cues are helpful in localizing objects outside a user's field of view, such as when a car comes around a blind corner in a driving simulation. They also help the separation of simultaneous sounds (e.g., many speakers at a cocktail party). Finally, qualitative changes in sound propagation, such as more absorption in a room with more plush carpets, can enhance and reinforce visual comprehension of the environment. Experiments have shown that more accurate acoustic modeling provides a user with a stronger sense of presence in a virtual environment.[2]

We have integrated our beam tracing method into an immersive system that allows a user to move through a virtual environment while images and spatialized audio are rendered in real-time according to the user's simulated viewpoint.[8–10] In the example shown in Fig. 23, multiple users represented by avatars (spheres) sharing a virtual world can speak to one another while the system spatializes their voices according to sound propagation paths (lines) through the environment.

In order to support multiple simultaneously moving sources and receivers,[9] as is required by a distributed virtual environment application with many avatars, we can no longer precompute beam trees. Instead, we must compute them in real-time as the source moves. However, we can take advantage of the fact that sounds can only be generated or heard at the positions of ''avatars'' representing the users. This simple observation enables two important enhancements to our beam tracing method. First, a bidirectional beam tracing algorithm combines beams traced from both sources and receivers to find propagation paths between them. Second, an amortized beam tracing algorithm computes beams emanating from box-shaped regions of space containing predicted avatar locations and reuses those beams multiple times to compute propagation paths as each avatar moves inside the box. We have incorporated these two enhancements into a time-critical multiprocessing system that allocates its computational resources dynamically in order to compute the highest priority propagation paths between moving avatar locations in real-time with graceful degradation and adaptive refinement. These enhancements result in two orders of magnitude of improvement in computational efficiency in the case where beams are traced for known receiver locations. See Ref. 9 for details.

Overall, we find that precomputing beams is advantageous for stationary sound sources and arbitrarily moving receivers, while computing them asynchronously on the fly is still practical for continuously moving sources and receivers.

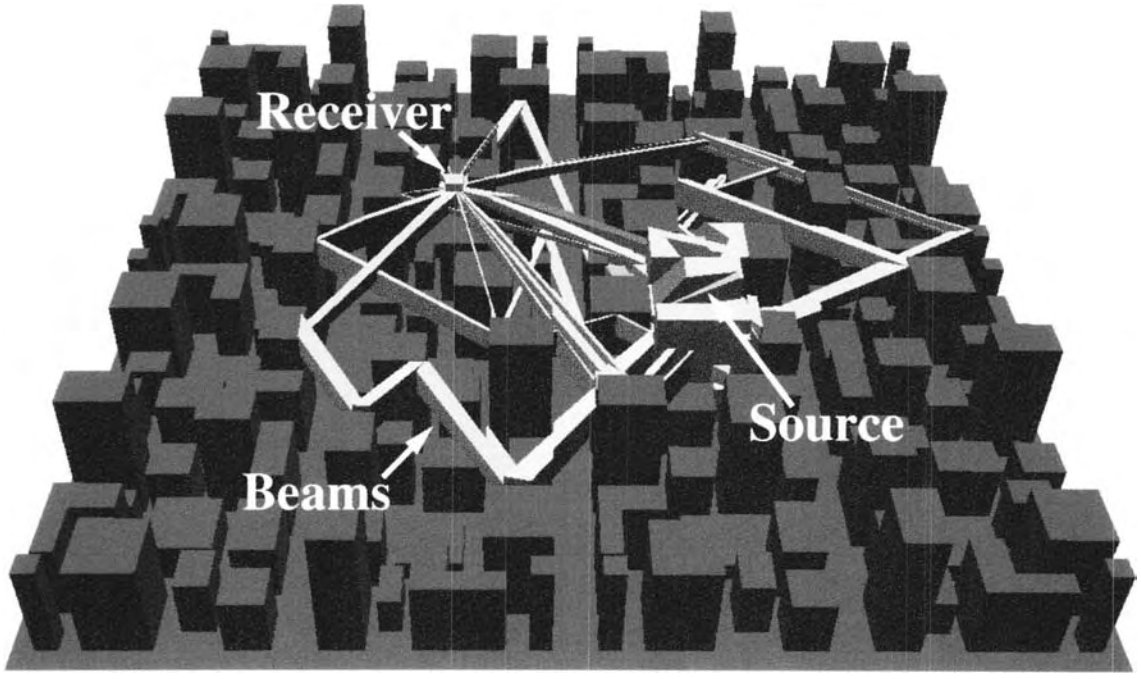


FIG. 25. Beams (green) containing all eighth-order specular reflection paths from a source to a receiver in City model.

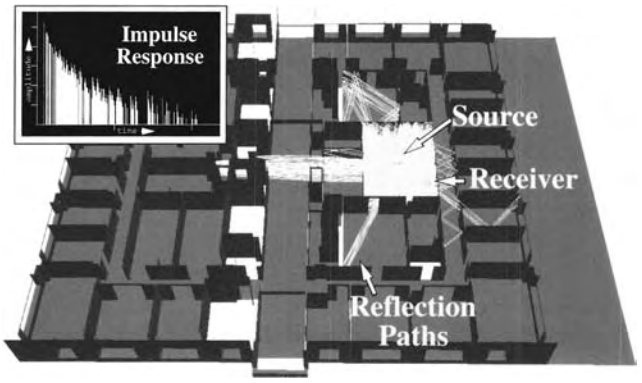


FIG. 26. Impulse response (inset) derived from eighth-order specular reflection paths (yellow) in Floor model.

B. Visualization

In order to aid the understanding and debugging of our acoustic modeling method, we find it extremely valuable to use interactive visualization of our data structures and algorithms. Our system provides menu and keyboard commands that may be used to toggle display of the (1) input polygons, (2) source point, (3) receiver point, (4) boundaries of the spatial subdivision, (5) pyramidal beams, (6) image sources, and (7) propagation paths. The system also supports visualization of acoustic metrics (e.g., power, clarity, etc.) for a set of receiver locations on a regular planar grid displayed with a textured polygon. Example visualizations are shown in Figs. 24–26.

Of course, many commercial[11–13] and research systems[29,71] provide elaborate tools for visualizing computed acoustic metrics. The critical difference in our system is that it supports continuous interactive updates of propagation paths and debugging information as a user moves the receiver point with the mouse. For instance, Figs. 24 and 26 show eighth-order specular reflection paths from a single audio source to a receiver location which is updated more than six times per second as the receiver location is moved arbitrarily. Figure 27 shows paths with specular reflections and diffractions computed in a city model and an auditorium. The user may select any propagation path for further inspection by clicking on it and then independently toggle display of reflecting cell boundaries, transmitting cell boundaries, and the polyhedral beams associated with the selected path.

Separate pop-up windows provide real-time display of other useful visual debugging and acoustic modeling information. For instance, one window shows a diagram of the beam tree data structure. Each beam tree node is dynamically colored in the diagram according to whether the receiver point is inside its associated beam or cell. Another window shows a plot of the impulse response representing the propagation paths from source to receiver (see Fig. 26). A third window shows values of various acoustic metrics, including power, clarity, reverberation time, and frequency response. All of the information displayed is updated in real-time as the user moves the receiver interactively with the mouse.

C. Geometric limitations

Our system is a research prototype, and it has several limitations. First, the 3D model must comprise only planar polygons because we do not model the transformations for beams as they reflect off curved surfaces. Furthermore, we do not trace beams along paths of refraction or diffuse reflection, which may be important acoustical effects. Each acoustic reflector is assumed to be locally reacting and to have dimensions far exceeding the wavelength of audible sound.

Second, our methods are only practical for coarse 3D models without highly faceted surfaces, such as the ones often found in acoustic modeling simulations of architectural spaces and concert halls. The difficulty is that beams are fragmented by cell boundaries as they are traced through a cell adjacency graph. For this reason, our beam tracing method would not perform well for geometric models with high local geometric complexity (e.g., a forest of trees).

Third, the major occluding and reflecting surfaces of the virtual environment must be static through the entire execution. If any acoustically significant polygon were to move, the cell adjacency graph would have to be updated incrementally.

The class of geometric models for which our method does work well includes most architectural and urban environments. In these cases, acoustically significant surfaces are generally planar, large, and stationary, and the acoustical effects of any sound source are limited to a local region of the environment (*densely occluded*).

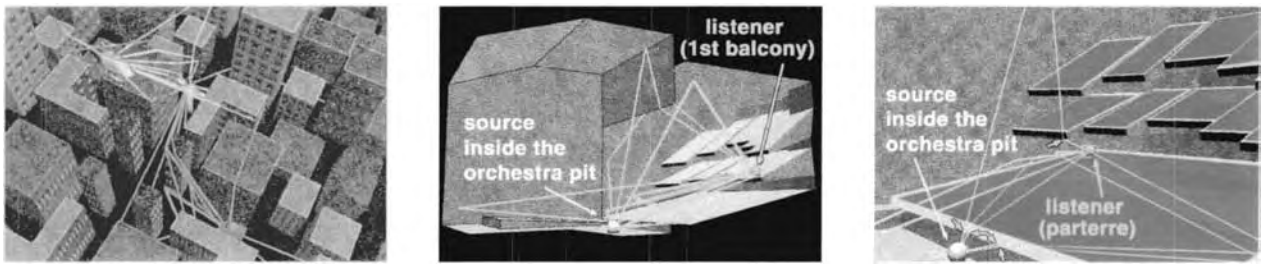


FIG. 27. Visualizations of sound paths in different environments. Diffraction of sound in a city environment is shown on the left, while early propagation paths for a source located in the orchestra pit of an opera house are shown in the middle and right images. Note the diffracted paths over the lip of the pit and balconies (cyan arrows).

D. Future work

Our system could be extended in many ways. For instance, the beam tracing algorithm is well suited for parallelization, with much of the previous work in parallel ray tracing directly applicable.[72] Also, the geometric regions covered by each node of the beam tree could be stored in a single hierarchical spatial structure (e.g., a BSP), allowing logarithmic search during path generation, rather than linear search of the beams inside a single cell. Of course, we could also use beam trees to allow a user to manipulate the acoustic properties of individual surfaces of the environment interactively with real-time feedback, such as for parametrized ray tracing[73] or for inverse modeling.[74]

Verification of our simulation results by comparison to measured data is an important topic for further study. In this paper, we have purposely focused on the computational aspects of geometrical acoustic modeling and left the validation of the models for future work. For this aspect of the problem, we refer the reader to related verification studies (e.g., Ref. 75), noting that our current system can compute the same specular reflection paths as methods based on image sources and ray tracing for which verification results are published (e.g., Refs. 76 and 77).

We are currently making impulse response measurements for verification of our simulations with reflections, diffractions, and transmissions. We have recently built a ''room'' for validation experiments. The base configuration of the room is a simple box. However, it is constructed with reconfigurable panels that can be removed or inserted to create a variety of interesting geometries, including ones with diffracting panels in the room's interior. We are currently measuring the directional reflectance distribution of each of these panels in the Anechoic Chamber at Bell Laboratories. We plan to make measurements with speakers and microphones at several locations in the room and with different geometric arrangements of panels, and we will compare the measurements with the results of simulations with the proposed beam tracing algorithms for matching configurations.

Perhaps the most interesting direction of future work is to investigate the possible applications of *interactive* acoustic modeling. What can we do with interactive manipulation of acoustic model parameters that would be difficult to do otherwise? As a first application, we hope to build a system that uses our interactive acoustic simulations to investigate the psychoacoustic effects of varying different acoustic modeling parameters. Our system will allow a user to interactively change various acoustic parameters with real-time auralization and visualization feedback. With this interactive simulation system, it may be possible to address psychoacoustic questions, such as ''how many reflections are psychoacoustically important to model?'' or ''which surface reflection model provides a psychoacoustically better approximation?'' Moreover, we hope to investigate the interaction of visual and aural cues on spatial perception. We believe that the answers to such questions are of critical importance to future design of 3D simulation systems.

VII. CONCLUSION

We have described a system that uses beam tracing data structures and algorithms to compute early propagation paths from static sources to a moving receiver at interactive rates for real-time auralization in large architectural environments.

As compared to previous acoustic modeling approaches, our beam tracing method takes unique advantage of *precomputation* and *convexity*. Precomputation is used twice, once to encode in the spatial subdivision data structure a depth-ordered sequence of (cell boundary) polygons to be considered during any traversal of space, and once to encode in the beam tree data structure the region of space reachable from a static source by sequences of specular reflections, diffractions, and transmissions at cell boundaries. We use the convexity of the beams, cell regions, and cell boundary polygons to enable efficient and robust computation of beam-polygon and beam-receiver intersections. As a result, our method is uniquely able to (1) enumerate all propagation paths robustly, (2) scale to compute propagation paths in large, densely occluded environments, (3) model effects of edge diffraction in arbitrary polyhedral environments, and (4) support evaluation of propagation paths at interactive rates. Our interactive system integrates real-time auralization with visualization of large virtual environments.

Based on our initial experiences with this system, we believe that interactive geometric acoustic modeling provides a valuable new tool for understanding sound propagation in complex 3D environments. We are continuing this research in order to further investigate the perceptual interaction of visual and acoustical effects and to better realize the opportunities possible with interactive acoustic modeling.

ACKNOWLEDGMENTS

The authors thank Sid Ahuja for his support of this work, and Arun C. Surendran and Michael Gatlin for their valuable discussions and contributions to the project.

[1] D. R. Begault, *3D Sound for Virtual Reality and Multimedia* (Academic, New York, 1994).
[2] N. I. Durlach and A. S. Mavor, *Virtual Reality Scientific and Technological Challenges*, National Research Council Report (National Academy, Washington, D.C., 1995).
[3] H. Kuttruff, *Room Acoustics (3rd edition)* (Elsevier Applied Science, New York, 1991).
[4] J. B. Allen and D. A. Berkley, ''Image method for efficiently simulating small room acoustics,'' J. Acoust. Soc. Am. **65**, 943–950 (1979).
[5] J. Borish, ''Extension of the image model to arbitrary polyhedra,'' J. Acoust. Soc. Am. **75**, 1827–1836 (1984).
[6] U. R. Krockstadt, ''Calculating the acoustical room response by the use of a ray tracing technique,'' J. Sound Vib. **8**(18), 118–125 (1968).
[7] M. Kleiner, B. I. Dalenback, and P. Svensson, ''Auralization—An overview,'' J. Audio Eng. Soc. **41**(11), 861–875 (1993).
[8] T. Funkhouser, I. Carlbom, G. Elko, G. Pingali, M. Sondhi, and J. West, ''A beam tracing approach to acoustic modeling for interactive virtual environments,'' *ACM Computer Graphics, SIGGRAPH'98 Proceedings*, July 1998, pp. 21–32.
[9] T. Funkhouser, P. Min, and I. Carlbom, ''Real-time acoustic modeling for distributed virtual environments,'' *ACM Computer Graphics, SIGGRAPH'99 Proceedings*, August 1999, pp. 365–374.
[10] N. Tsingos, T. Funkhouser, A. Ngan, and I. Carlbom, ''Modeling acoustics in virtual environments using the uniform theory of diffraction,'' *ACM Computer Graphics, SIGGRAPH 2001 Proceedings*, August 2001, pp. 545–552.

[11] Bose Corporation, Bose Modeler, Framingham, MA, http://www.bose.com.
[12] CATT-Acoustic, Gothenburg, Sweden, http://www.netg.se/catt.
[13] J. M. Naylor, ''Odeon—Another hybrid room acoustical model,'' Appl. Acoust. **38**(1), 131–143 (1993).
[14] R. D. Ciskowski and C. A. Brebbia (eds.), *Boundary Element Methods in Acoustics* (Elseiver Applied Science, New York, 1991).
[15] P. Filippi, D. Habault, J. P. Lefevre, and A. Bergassoli, *Acoustics, Basic Physics, Theory and Methods* (Academic, New York, 1999).
[16] A. Kludszuweit, ''Time iterative boundary element method (TIBEM)—a new numerical method of four-dimensional system analysis for the calculation of the spatial impulse response,'' Acustica **75**, 17–27 (1991) (in German).
[17] S. Kopuz and N. Lalor, ''Analysis of interior acoustic fields using the finite element method and the boundary element method,'' Appl. Acoust. **45**, 193–210 (1995).
[18] G. R. Moore, ''An Approach to the Analysis of Sound in Auditoria,'' Ph.D. thesis, Cambridge, UK, 1984.
[19] N. Tsingos and J.-D. Gascuel, ''Soundtracks for computer animation: Sound rendering in dynamic environments with occlusions,'' *Graphics Interface'97*, May 1997, pp. 9–16.
[20] M. F. Cohen and J. R. Wallace, *Radiosity and Realistic Image Synthesis* (Academic, New York, 1993).
[21] F. X. Sillion and C. Puech, *Radiosity and Global Illumination* (Morgan Kaufmann, San Francisco, 1994).
[22] K. H. Kuttruff, ''Auralization of impulse responses modeled on the basis of ray-tracing results,'' J. Audio Eng. Soc. **41**(11), 876–880 (1993).
[23] U. R. Kristiansen, A. Krokstad, and T. Follestad, ''Extending the image method to higher-order reflections,'' J. Appl. Acoust. **38**(2-4), 195–206 (1993).
[24] R. Cook, T. Porter, and L. Carpenter, ''Distributed ray-tracing,'' *ACM Computer Graphics, SIGGRAPH'84 Proceedings*, July 1984, Vol. 18(3), pp. 137–146.
[25] J. T. Kajiya, ''The rendering equation,'' *ACM Computer Graphics, SIGGRAPH'86 Proceedings*, Vol. 20(4), pp. 143–150.
[26] H. Lehnert, ''Systematic errors of the ray-tracing algorithm,'' Appl. Acoust. **38**, 207–221 (1993).
[27] N. Dadoun, D. G. Kirkpatrick, and J. P. Walsh, ''The geometry of beam tracing,'' *Proceedings of the Symposium on Computational Geometry*, June 1985, pp. 55–71.
[28] P. Heckbert and P. Hanrahan, ''Beam tracing polygonal objects,'' *ACM Computer Graphics, SIGGRAPH'84 Proceedings*, Vol. 18(3), pp. 119–127.
[29] M. Monks, B. M. Oh, and J. Dorsey, ''Acoustic simulation and visualisation using a new unified beam tracing and image source approach,'' *Proc. Audio Engineering Society Convention*, 1996, pp. 153–174.
[30] U. Stephenson and U. Kristiansen, ''Pyramidal beam tracing and time dependent radiosity,'' *Fifteenth International Congress on Acoustics*, June 1995, pp. 657–660.
[31] J. P. Walsh and N. Dadoun, ''What are we waiting for? The development of Godot, II,'' 103rd Meeting of the Acoustical Society of America, April 1982.
[32] J. H. Chuang and S. A. Cheng, ''Computing caustic effects by backward beam tracing,'' Visual Comput. **11**(3), 156–166 (1995).
[33] A. Fujomoto, ''Turbo beam tracing—A physically accurate lighting simulation environment,'' *Knowledge Based Image Computing Systems*, May 1988, pp. 1–5.
[34] G. Ghazanfarpour and J. M. Hasenfratz, ''A beam tracing with precise antialiasing for polyhedral scenes,'' Comput. Graph. **22**(1), 103–115 (1998).
[35] E. Haines, ''Beams O' Light: Confessions of a hacker,'' *Frontiers in Rendering, Course Notes, SIGGRAPH'91*, 1991.
[36] M. Watt, ''Light-water interaction using backward beam tracing,'' *ACM Computer Graphics, SIGGRAPH'90 Proceedings*, August 1990, pp. 377–385.
[37] C. B. Jones, ''A new approach to the 'hidden line' problem,'' Comput. J. **14**(3), 232–237 (1971).
[38] T. Funkhouser, ''A visibility algorithm for hybrid geometry- and image-based modeling and rendering,'' Comput. Graph. **23**(5), 719–728 (1999).
[39] S. Teller, ''Visibility Computations in Densly Occuded Polyhedral Environments,'' Ph.D. thesis, Computer Science Div., University of California, Berkeley, 1992.
[40] S. Fortune, ''Algorithms for prediction of indoor radio propagation,'' Technical Report Document 11274-960117-03TM, Bell Laboratories, 1996.
[41] S. J. Fortune, ''Topological beam tracing,'' in *Proc. 15th ACM Symposium on Computational Geometry*, 1999, pp. 59–68.
[42] J. Amanatides, ''Ray tracing with cones,'' *ACM Computer Graphics, SIGGRAPH'84 Proceedings*, July 1984, Vol. 18(3), pp. 129–135.
[43] J. P. Vian and D. van Maercke, ''Calculation of the room response using a ray tracing method,'' *Proceedings of the ICA Symposium on Acoustics and Theater Planning for the Performing Arts*, 1986, pp. 74–78.
[44] T. Lewers, ''A combined beam tracing and radiant exchange computer model of room acoustics,'' Appl. Acoust. **38**, 161–178 (1993).
[45] P. Kreuzgruber, P. Unterberger, and R. Gahleitner, ''A ray splitting model for indoor radio propagation associated with complex geometries,'' in *Proceedings of the 1993 43rd IEEE Vehicular Technology Conference*, 1993, pp. 227–230.
[46] A. Rajkumar, B. F. Naylor, F. Feisullin, and L. Rogers, ''Predicting RF coverage in large environments using ray-beam tracing and partitioning tree represented geometry,'' Wireless Networks **2**(2), 143–154 (1996).
[47] S. Teller and P. Hanrahan, ''Global visibility algorithms for illumination computations,'' pp. 239–246 (1993).
[48] S. Teller, C. Fowler, T. Funkhouser, and P. Hanrahan, ''Partitioning and ordering large radiosity computations,'' *ACM Computer Graphics, SIGGRAPH'93 Proceedings*, August 1994, pp. 443–450.
[49] P. Min and T. Funkhouser, ''Priority-driven acoustic modeling for virtual environments,'' *EUROGRAPHICS 2000*, August 2000, pp. 179–188.
[50] F. Durand, G. Drettakis, and C. Puech, ''The visibility skeleton: A powerful and efficient multi-purpose global visibility tool,'' in *Computer Graphics, SIGGRAPH'97 Proceedings*, 1997, pp. 89–100.
[51] B. G. Baumgart, ''Winged edge polyhedron representation,'' Technical Report AIM-179 (CS-TR-74-320), Computer Science Department, Stanford University, Palo Alto, CA, October 1972.
[52] D. P. Dobkin and M. J. Laszlo, ''Primitives for the manipulation of three-dimensional subdivisions,'' Algorithmica **4**(1), 3–32 (1989).
[53] H. Fuchs, Z. M. Kedem, and B. F. Naylor, ''On visible surface generation by a priori tree structures,'' *ACM Computer Graphics, SIGGRAPH '80 Proceedings*, July 1980, Vol. 14(3), pp. 124–133.
[54] B. F. Naylor, ''Constructing good partitioning trees,'' *Graphics Interface'93*, May 1993, pp. 181–191.
[55] J. B. Keller, ''Geometrical theory of diffraction,'' J. Opt. Soc. Am. **52**(2), 116–130 (1962).
[56] S. Teller, ''Computing the antiumbra cast by an area light source,'' *ACM Computer Graphics, SIGGRAPH'92 Proceedings*, July 1992, Vol. 26(2), pp. 139–148.
[57] J. Goodman and J. O'Rourke (eds.), *Handbook of Discrete and Computational Geometry* (CRC, Boca Raton, FL, 1997).
[58] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in C, 2nd edition* (Cambridge U.P., New York, 1992).
[59] L. Aveneau, Y. Pousset, R. Vauzelle, and M. Méria ux, ''Development and evaluations of physical and computer optimizations for the 3d utd model,'' *AP2000 Millennium Conference on Antennas & Propagation* (Poster), April 2000.
[60] H. Lehnert and J. Blauert, ''Principles of binaural room simulation,'' Appl. Acoust. **36**, 259–291 (1992).
[61] M. E. Delany and E. N. Bazley, ''Acoustical characteristics of fibrous absorbent materials,'' Technical Report NPL AERO REPORT Ac37, National Physical Laboratory, Aerodynamics Division, March 1969.
[62] A. D. Pierce, *Acoustics. An introduction to its physical principles and applications*, 3rd ed. (American Institute of Physics, New York, 1984).
[63] S.-I. Thomasson, ''Reflection of waves from a point source by an impedance boundary,'' J. Acoust. Soc. Am. **59**, 780–785 (1976).
[64] R. G. Kouyoumjian and P. H. Pathak, ''A uniform geometrical theory of diffraction for an edge in a perfectly conducting surface,'' Proc. IEEE **62**, 1448–1461 (1974).
[65] T. Kawai, ''Sound diffraction by a many sided barrier or pillar,'' J. Sound Vib. **79**(2), 229–242 (1981).
[66] M. A. Biot and I. Tolstoy, ''Formulation of wave propagation in infinite media by normal coordinates with an application to diffraction,'' J. Acoust. Soc. Am. **29**, 381–391 (1957).
[67] H. Medwin, E. Childs, and G. Jebsen, ''Impulse studies of double diffraction: A discrete huygens interpretation,'' J. Acoust. Soc. Am. **72**, 1005–1013 (1982).
[68] U. P. Svensson, R. I. Fred, and J. Vanderkooy, ''Analytic secondary source model of edge diffraction impulse responses,'' J. Acoust. Soc. Am. **106**, 2331–2344 (1999).

[69] R. R. Torres, U. P. Svensson, and M. Kleiner, ''Computation of edge diffraction for more accurate room acoustics auralization,'' J. Acoust. Soc. Am. **109**, 600–610 (2001).
[70] J. Blauert, *Spatial Hearing: The Psychophysics of Human Sound Localization* (MIT, Cambridge, MA, 1983).
[71] A. Stettner and D. P. Greenberg, ''Computer graphics visualization for acoustic simulation,'' *ACM Computer Graphics, SIGGRAPH'89 Proceedings*, July 1989, Vol. 23(3), pp. 195–206.
[72] J. Arvo and D. Kirk, ''A survey of ray tracing acceleration techniques,'' *An Introduction to Ray Tracing*, 1989.
[73] C. Sequin and E. Smyrl, ''Parameterized ray tracing,'' *ACM Computer Graphics, SIGGRAPH'89 Proceedings*, July 1989, Vol. 23(3), pp. 307–314.
[74] M. Monks, B. M. Oh, and J. Dorsey, ''Audioptimization: Goal based acoustic design,'' *IEEE Computer Graphics & Applications*, May 2000, pp. 76–91.
[75] M. Vorlander, ''International round robin on room acoustical computer simulations,'' in *Proceedings of the 15th International Congress of Acoustics*, June 1995.
[76] K. Nakagawa, T. Miyajima, and Y. Tahara, ''An improved geometrical sound field analysis in rooms using scattered sound and an audible room acoustic simulator,'' J. Appl. Acoust. **38**(2-4), 115–130 (1993).
[77] G. M. Naylor and J. H. Rindel, ''Predicting room acoustical behavior with the odeon computer model,'' in *Proceedings of the 124th ASA Meeting*, November 1992, p. 3aAA3.

Modeling Acoustics in Virtual Environments Using the Uniform Theory of Diffraction

Nicolas Tsingos[1], Thomas Funkhouser[2], Addy Ngan[2], Ingrid Carlbom[1]

[1] Bell Laboratories*

[2] Princeton University†

Abstract

Realistic modeling of reverberant sound in 3D virtual worlds provides users with important cues for localizing sound sources and understanding spatial properties of the environment. Unfortunately, current geometric acoustic modeling systems do not accurately simulate reverberant sound. Instead, they model only direct transmission and specular reflection, while diffraction is either ignored or modeled through statistical approximation. However, diffraction is important for correct interpretation of acoustic environments, especially when the direct path between sound source and receiver is occluded.

The Uniform Theory of Diffraction (UTD) extends geometrical acoustics with diffraction phenomena: illuminated edges become secondary sources of diffracted rays that in turn may propagate through the environment. In this paper, we propose an efficient way for computing the acoustical effect of diffraction paths using the UTD for deriving secondary diffracted rays and associated diffraction coefficients. Our main contributions are: 1) a beam tracing method for enumerating sequences of diffracting edges efficiently and without aliasing in densely occluded polyhedral environments; 2) a practical approximation to the simulated sound field in which diffraction is considered only in shadow regions; and 3) a real-time auralization system demonstrating that diffraction dramatically improves the quality of spatialized sound in virtual environments.

Keywords: Spatialized Sound, Virtual Environments, Sound Visualization, Uniform Theory of Diffraction, Beam Tracing.

1 Introduction

Realistic simulation of virtual environments has been a major focus of research in interactive computer graphics for decades, dating back to the early flight simulators of the 1960s. Most prior research focused on visualization, while relatively little attention was paid to auralization. However, auditory cues are important in immersive virtual environments, as they combine with visual cues to aid in localization of objects, separation of simultaneous sound signals, and formation of spatial impressions [5] which enhance and reinforce the visual comprehension of the environment. Experiments have shown that accurate acoustic modeling gives a user a stronger sense of presence in a virtual environment [9], and that high-quality audio enhances perceived visual quality [37].

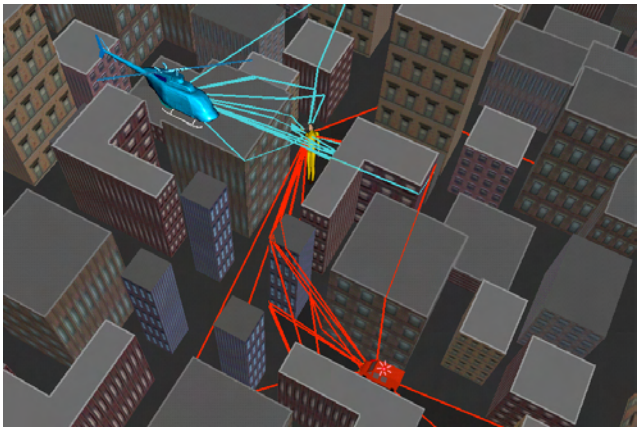


Figure 1: Early diffracted and reflected sound paths in a city environment where direct sound from sources is occluded. We use the Uniform Theory of Diffraction which considers edges in the environment as sources of new diffracted rays, complementing reflected and transmitted rays.

Current virtual environment systems render audio using geometrical techniques, such as image-sources, ray-tracing or beam-tracing, to compute early propagation paths between sound sources and listener while late reverberation is usually modeled using statistical techniques [12, 13, 34]. Unfortunately, they fail to spatialize sound realistically because they do not accurately account for diffraction. Our goal is to compute early geometrical propagation paths, including diffraction effects, that can be used for real-time auralization in such a system.

Diffraction is a form of scattering by obstacles whose size is of the same order of magnitude as the wavelength. It is a fundamental mode of sound propagation, particularly in building interiors and cities where the direct path between a sound source and a receiver is often occluded. For example, consider the training simulation scenario shown in Figure 1, in which a pedestrian must respond to a helicopter and a fire engine. Failure to simulate sound diffraction could be "disastrous," since the person cannot see either sound source, and the direct path from both sound sources is blocked by tall buildings. In such situations, it is important that the virtual environment system simulates diffraction correctly because people localize sounds by interpreting echoes according to their delays relative to the first arriving wavefront [5, 30]. Moreover, if diffraction is omitted from a virtual environment simulation, the user may experience abrupt changes in spatialized sounds as he/she turns a corner and the sound source disappears from the line of sight. For instance, consider walking down the hallway of your office building and having the sound disappear after you pass each open door.

*{tsingos|carlbom}@research.bell-labs.com

†{funk|waingan}@cs.princeton.edu

Such abrupt changes in a simulation would introduce a mismatch with our real world experiences, which would result in "negative training" or at least confuse users.

In this paper, we propose an efficient way for computing the acoustical effect of early reflection and diffraction paths according to the Uniform Theory of Diffraction [21, 24, 26]. Specifically, we make three contributions. First, we describe a beam tracing method, for enumerating sequences of diffracting edges efficiently and without aliasing in densely occluded polyhedral environments. Second, we propose an approximation to simulated sound fields suitable for immersive virtual environments in which diffraction is computed only in shadow regions. Finally, we describe a real-time auralization system that produces spatialized sound with early diffraction, transmission, and specular reflection during interactive walkthroughs of complex environments.

Our experimental results demonstrate that (1) beam tracing is an efficient and aliasing-free way to find diffraction sequences in densely occluded environments, (2) it is possible to construct early diffracting propagation paths and spatialize sounds in real-time, and (3) diffraction greatly improves the quality of spatialized sounds in immersive virtual environments.

2 Background and Related Work

There are currently three major approximation theories for diffraction problems in polyhedral environments: (1) the Huygens-Fresnel diffraction theory, (2) boundary integral representations using the Helmoltz-Kirchoff integral theorem, and (3) the Uniform Theory of Diffraction.

Huygens' principle [17] predicts that every point on a wavefront can be regarded as the source of a secondary spherical wavelet. The wavefield is defined at each point by the superposition of these wavelets, which extend in all directions, including shadow regions. Fresnel supplemented Huygens' theory by adding interference between the wavelets to treat diffraction [6]. He also subdivided space between the source and the receiver into concentric ellipsoids with frequency-dependent radii: the Fresnel ellipsoids. By modeling diffraction effects as a loss in signal intensity, Bertoni [4] and Tsingos and Gascuel [43] use Fresnel ellipsoids to determine relevant obstacles at any given frequency. By replacing the binary geometrical visibility by an extended visibility term between 0 and 1, they achieve frequency-dependent sound "muffling." This technique removes abrupt cuts in the simulated sound, producing a more pleasing experience. However, it fails to capture the temporal aspect of diffraction since new propagation paths are not introduced. While this approximation is not usually a concern for electromagnetic wave propagation, it is an important issue for acoustics.

Analytic expressions give time-domain diffraction filters for sequences of finite wedges based on a discrete Huygens interpretation [27, 38]. In this case, the edges are discretized into secondary point sources whose contributions must be summed to obtain the diffracted field. Such models prove very accurate for low order diffraction and have recently been used to assess audibility of diffraction in the case of a simple stage house [41]. However, it is unclear if the method can be applied in real-time since the edges must be discretized into a large number of point sources to compute the diffraction filters.

The Helmoltz-Kirchoff integral theorem provides a formalization of the Huygens-Fresnel principle [6, 10]. It expresses the scattered field at any point in space as a function of the field on the surface of the diffracting objects. Mathematically, it can be expressed as a surface integral and solved by numerical methods such as Boundary Element Methods (BEM) [16, 18] that discretize surfaces into patches. BEM allow for very accurate treatment of sound diffraction. But, they are far too compute intensive for interactive sound rendering over the whole audio spectrum and are mainly used for low frequencies (below 150Hz). In some cases the integral can be solved analytically [35], such as for height fields or periodic surfaces. However, neither of these cases usually applies to architectural models.

The Uniform Theory of Diffraction (UTD) [21, 24, 26] incorporates diffraction into the ray theory of light. The UTD treats an infinite wedge as a secondary source of diffracted waves that in turn can be reflected and diffracted before reaching the receiver. For a given point source and point receiver location, the diffraction of a wave over an infinite wedge is represented by a *single* ray whose contribution to the wave field is attenuated by a complex valued diffraction coefficient [24] (see Appendix A). For any sequence of diffracting edges, the ray follows the path satisfying Fermat's principle: if the propagation medium is homogeneous, the ray follows the shortest path from the source to the receiver, stabbing the diffracting edges. The UTD is a high frequency approximation and applies to infinite wedges, when the source and listener remain far from diffracting surfaces (compared to the wavelength). To date, the UTD has been applied successfully in several types of off-line simulations, including acoustical diffraction over solitary wedges [20], lighting effects in room-sized scenes [2], and radio frequency propagation in buildings and cities [33, 22]. For acoustic waves, the method has been validated down to 150Hz for a small combination of diffracting wedges [20, 38]. Validation of the approach for more complex situations has not yet been achieved.

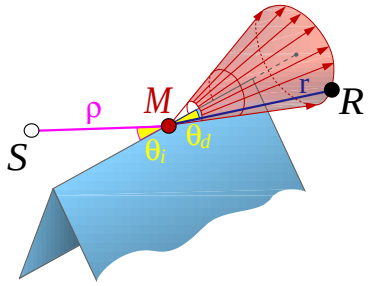
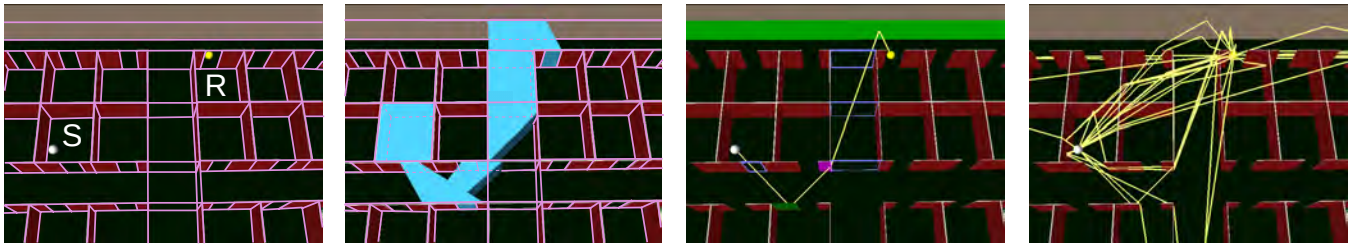


Figure 2: According to the UTD, an incoming ray ρ gives rise to a cone of diffracted rays, where the aperture angle θ_d of the cone is equal to the angle θ_i between the incident ray and the edge (the axis of the cone is the edge). For a given receiver location, a single ray describes the diffracted field.

Of these three approaches, the UTD is the most promising for spatializing sound in interactive virtual environments, as it integrates well into a geometrical framework, is physically-based, and provides satisfying results for most of the audio spectrum (for early diffraction orders).

The main computational challenge in using the UTD into real-time auralization systems is the efficient enumeration of significant early diffraction paths. Although many algorithms exist to find approximate solutions [29], they are either too inefficient or prone to aliasing. For instance, Aveneau [2] enumerated all permutations of polyhedral edges within the first few Fresnel ellipsoids, which is not practical for sound simulations in large environments. Rajkumar et al. [33] extended a ray tracing algorithm to broadcast rays in all directions for each edge "intersection." Similarly, Fortune et al. [11, 22] and Stephenson [36] described a beam tracing approach in which propagation due to each edge diffraction was approximated by a set of beams emanating from point sources at sampled positions along the diffracting edge. These latter two approaches approximate the set of potential diffraction paths by discrete sampling. Thus they are prone to aliasing, which would cause noticeable artifacts in an interactive sound system. Prior methods provide neither interactive response times nor guarantee finding all significant propagation paths.



(a) (b) (c) (d)

Figure 3: Overview of our process: (a) Virtual environment (office cubicles) with source S, receiver R, and spatial subdivision marked in pink. (b) Sample reflected and diffracted beam (cyan) containing the receiver. (c) Path generated for the corresponding sequence of faces (green), portals (purple), wedges (magenta). (d) The procedure repeated for all beams containing R.

In this paper, we describe real-time methods for simulating early sound reverberation with the Uniform Theory of Diffraction in large virtual environments. We address two main issues of such a system: (1) enumerating significant propagation paths efficiently, and (2) computing an approximation to the diffracted field that can be updated at interactive rates. Details are provided in the following sections.

3 Enumerating Propagation Paths

According to the UTD, an acoustic wave incident upon an edge between two non-coplanar surfaces forms a diffracted wave that propagates in a cone shaped pattern of rays from the intersected part of the edge, as shown in Figure 2. Our challenge is to represent the propagation pattern of these rays as they traverse free space, transmit through obstacles, and reflect off surfaces.

Our approach is based on object-precision beam tracing [15]. The motivation for this approach is to exploit the spatial coherence in propagation paths while avoiding the aliasing artifacts of sampling diffraction edges. In contrast to ray tracing, beam tracing works with object-precision polyhedral volumes that support well-defined intersections with diffracting edges. Aliasing resulting from the intersection of infinitely thin rays with infinitely thin edges is thus eliminated [33]. In contrast to brute-force enumeration of all edge permutations [2], beam tracing provides an effective method for pruning the search based on the feasibility of stabbing lines [39]. As a result, beam tracing finds every propagation path up to a specified termination criteria without undersampling errors. Moreover, beam tracing algorithms are practical for specular reflection in densely occluded virtual environments [12], and can be readily incorporated into interactive virtual environments systems [13].

In the following two subsections, we focus on the challenges of tracing beams and constructing propagation paths with diffraction. Unfortunately, as beams emanating from a source and diffracting over an edge are traced along subsequent sequences of reflections and transmissions, they can become quite complex, bounded by quadric surfaces due to triple-edge (EEE) events. Rather than representing these scattering patterns exactly [8, 40], we conservatively over-estimate the space of rays diffracting over an edge with a polyhedral approximation. We compensate for this approximation later by checking each propagation path to determine if it lies in the over-estimating part of the beam, in which case it is discarded. Since diffraction patterns are approximated conservatively and tightly with simple polyhedra, and checking propagation paths is quick, the whole process is much faster than computing the exact propagation pattern directly.

3.1 Beam Construction and Tracing

The goal is to enumerate the significant permutations of diffractions, specular reflections, and transmissions along which a sound wave can travel from a given source location. The algorithm must be conservative, so that no significant propagation paths are missed. But, it should not be too over-estimating, so that the second stage of our process becomes over-burdened with construction of infeasible propagation paths. Finally, to enable efficient checking of propagation paths, our algorithm must not only construct a beam containing the region of space reachable by each propagation sequence, but it must also encode potential blockers (or equivalently "portals").

We incrementally compute beams starting from a source by traversing the cell-face and face-edge adjacency graph of a polyhedral cell complex, as in [19, 1, 12, 13, 39] (see Figure 3). Starting in the cell containing the source with a beam representing the entire cell, we iteratively visit adjacent cells in priority order, considering different permutations of transmission, specular reflection, *and diffraction* resulting from the faces and edges on the boundary of the "current" cell. As each new cell C is visited, the current beam B is updated such that it contains all potential propagation paths along the current traversal sequence. We identify diffracting edges ε on the boundary of C as the ones: (1) intersected by B and (2) shared by two faces F_1 and F_2 on the boundary of C that are either non-coplanar or have different acoustic properties (e.g., F_1 is transparent and F_2 is opaque). For each such edge, we construct a new beam containing potential diffraction paths and begin tracing it through all adjacent cells. We also construct and trace beams for transparent and reflecting surfaces, as in [12]. All sequences and their corresponding beams are logged in a *beam tree* data structure [12, 15], which can be queried later to determine the set of propagation paths reaching a specific receiver location.

Each beam emanating from a diffracting edge and passing through or reflecting off a sequence of cell boundaries is represented conservatively by the intersection of two cones and a polytope (see Figure 4). The two cones are constructed with axes along the diffracting edge, with apexes at the two endpoints of the beam-edge intersection. Their interior angles are derived from the equal angle constraints at these endpoints, as dictated by the Uniform Theory of Diffraction. The polytope bounds the set of lines emanating from the diffracting edge and stabbing the traversed sequence of convex cell boundaries with a constant number of opposing planes, as described in [40]. This representation allows every beam traced through a sequence of arbitrarily oriented faces to keep a bounded complexity, which is important for both computational efficiency and memory utilization. Accordingly, using the adjacency information in the cell complex, each beam is updated incrementally in expected-case constant time.

Although there are generally exponentially many distinct sequences of diffraction through a 3D polyhedral scene, we expect a large number of them to be psychoacoustically insignificant (the amplitude of diffracted contributions quickly drops with the diffraction order), and thus beams are traced in priority order [28], either during an off-line precomputation (as in [12]) or in real-time using multiple asynchronous processes (as in [13]).

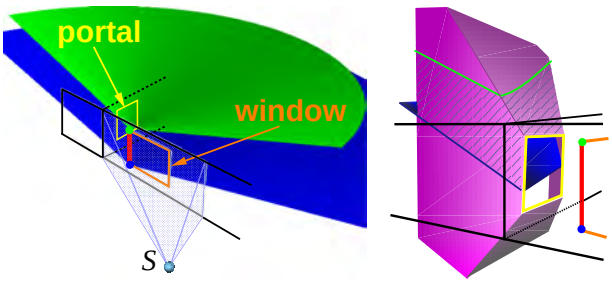


Figure 4: Possible diffraction paths are conservatively bounded by the intersection of two cones and a polytope. Left: a beam incident on the edge (red) of a window and the two diffraction cones (shown in blue and green only in the halfspace behind the window) computed at the endpoints of the beam-edge intersection. Right: a close-up view of the intersection (hatched) between the two cones and the polytope resulting from stepping through the next portal (yellow).

3.2 Path Construction and Validation

Once beams are traced from a sound source, we construct a unique propagation path for each beam containing a receiver location. The geometry of the path determines the delay, amplitude, and directivity of the sound wave traveling along the path from the source to the receiver.

According to the UTD, the wave field resulting from a diffraction can be approximated by a piecewise-linear propagation path – i.e., the shortest among all possible paths from the source to the receiver stabbing the faces and edges in the sequence. In order to construct this path for a given sequence of beams, we find the points of intersection for every reflecting face and diffracting edge. The intersections with specularly reflecting faces are uniquely determined by the locations of the source, receiver and diffraction points. Thus, the problem is reduced to finding the locations of the diffraction points, P_i $(i = 1 \ldots n)$ (see Figure 5). At each of these points, the path satisfies a simple "unfolding property" (see Figure 2): the angle (θ_i) at which the path enters the edge must be the same as the angle (θ_d) at which it leaves [14]. Thus, for each potential path, we solve a non-linear system of n equations expressing equal angle constraints at the diffracting edges:

$$\begin{cases} \overrightarrow{P_1S} \cdot \overrightarrow{E_1} & = & \overrightarrow{P_1P_2} \cdot (-\overrightarrow{E_1}) \\ \overrightarrow{P_2P_1} \cdot \overrightarrow{E_2} & = & \overrightarrow{P_2P_3} \cdot (-\overrightarrow{E_2}) \\ & \vdots & \\ \overrightarrow{P_nP_{n-1}} \cdot \overrightarrow{E_n} & = & \overrightarrow{P_nR} \cdot (-\overrightarrow{E_n}) \end{cases} \quad (1)$$

where S is the source point, R is the receiver point, $\overrightarrow{E_i}$ is the normalized direction vector of the ith diffracting edge, and $\overrightarrow{P_{i+1}P_i}$ is a normalized direction vector between two adjacent points in the shortest path. To incorporate specular reflections in this equation, $\overrightarrow{E_i}$ and $\overrightarrow{P_{i+1}P_i}$ are transformed by successive mirroring operators accounting for the sequence of specularly reflecting faces up to the ith diffraction.

Parameterizing the edges, $P_i = O_i + t_i\overrightarrow{E_i}$ (where O_i is a reference point on edge i), the system of equations (1) is rewritten in terms of n unknowns t_i and solved within a specified tolerance using a non-linear system solving scheme. We use a locally convergent Newton scheme [32], with the middle of the edges as the initial estimate for the diffraction points. Since the equation satisfied by any diffraction point only depends on the previous and next

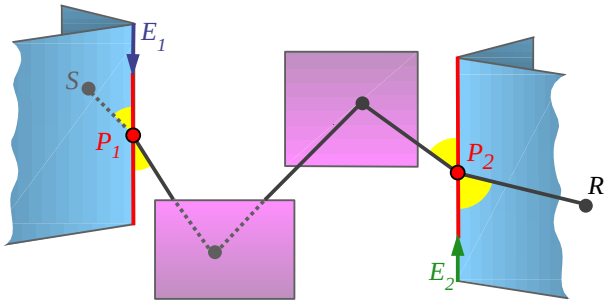


Figure 5: A single propagation path comprising a diffraction, two specular reflections, and another diffraction. The two diffraction points (P_i) are determined by equal angle constraints at the corresponding edges (E_i).

diffraction points in the sequence, the Jacobian matrix is tridiagonal and is easily evaluated analytically. Thus, every Newton iteration is performed in time $O(n)$ where n is the number of unknowns (i.e., edges). We found this method faster than the recursive construction proposed by Aveneau [3].

Once the diffraction points are found, we validate whether the resulting path intersects every cell boundary in the sequence (to compensate for the fact that the beams are conservatively approximate). If not, the path in the over-estimating part of the beam is discarded. Otherwise, it contributes to an *impulse response* used for spatializing sounds [23, 25] (see Appendix A).

The proposed conservative beam tracing and path construction enumerate all sequences of diffracting edges (without aliasing) up to a specified termination criterion, while most acoustically infeasible sequences of edges and faces are culled already during beam tracing.

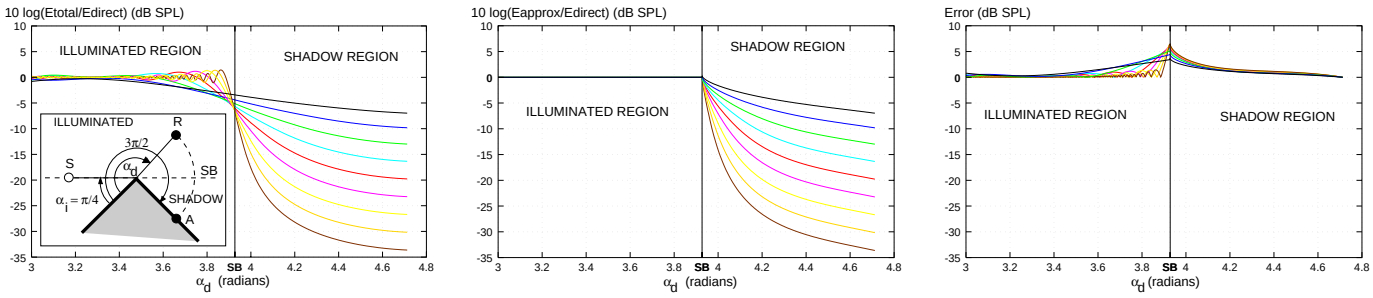
4 Shadow Region Approximation

Our beam tracing technique provides a method for finding diffraction paths efficiently and without aliasing. However, contrary to specular reflections or transmissions, diffraction introduces a scattering in all directions around the wedge, which results in a combinatorial explosion of the number of beams to consider, even for moderately complex scenes. In this section we introduce an approximation to reduce the spatial extent of diffraction beams, while preserving a good modeling of the diffracted field. This approximation enables us to achieve interactive auralization in large environments.

Recent psychoacoustic tests in the context of a simple stage house model [41] show that diffractions can be perceived in illuminated regions where direct and reflected contributions from a source also reach the listener. However, in this case, we conjecture that diffraction does not modify the main acoustic cues already carried by the direct and reflected contributions since the amplitude of the diffracted field is usually much weaker than direct or even reflected contributions [31] (p.500). On the other hand, we note that diffraction into shadow regions is crucial for typical virtual worlds as it provides the primary mode of propagation to most of the environment. Thus, we introduce an approximation in which *the contribution of diffraction is considered only in shadow regions*.

Accordingly, our current on-line implementation allows for adding an extra halfspace to the polytope representing each diffraction beam so that it tightly, yet conservatively, bounds the shadow region of each diffraction.

However, discarding the diffracted field in the illuminated region of a wedge introduces a discontinuity at the shadow boundary, as the direct field is abruptly replaced by the diffracted field. This is due to the fact that the UTD diffracted field is defined to ensure that the *sum* of the direct and diffracted fields is continuous for any



(a) UTD total wave field (b) Approximation (c) Error

Figure 6: Plots of (a) the UTD total wave field, (b) our approximation, and (c) the error as a function of diffraction angle (α_d), as the receiver rotates around the edge, for a single diffracting wedge (inset). Each plot shows several curves corresponding to the sound pressure level (SPL) for the center frequencies of octave bands ranging from 100 Hz (top) to 24kHz (bottom). Our approximation culls the diffracted field contribution in the illuminated region of the wedge but still closely matches the original UTD field.

receiver location (see Figure 6(a)), while both fields, independently, are discontinuous at shadow boundaries.

To preserve continuity at shadow boundaries, we normalize the diffracted field as predicted by the UTD, so that it is C^0 continuous with the direct field at the shadow boundary, *SB*. We define the normalized diffracted field at the receiver location R as:

$$\mathcal{E}'_{\text{diffracted}}(R) = \mathcal{E}^{\text{SB}}_{\text{incident}}(R)\mathcal{E}_{\text{diffracted}}(R)/\mathcal{E}^{\text{SB}}_{\text{diffracted}}(R), \quad (2)$$

where $\mathcal{E}^{\text{SB}}_{\text{incident}}(R)$ and $\mathcal{E}^{\text{SB}}_{\text{diffracted}}(R)$ are the incident and diffracted fields when the receiver R is rotated to lie on the shadow boundary SB (at the same distance from the edge).

This modified expression scales the diffracted field equally for all directions around the edge, unnecessarily modifying the original UTD field away from the shadow boundary. Hence, our new approximated diffracted field is derived by interpolating between the expression given by equation (2) and the original UTD expression of the diffracted field [24] (see Appendix A) as the receiver further moves inside the shadow region (between SB and A in Figure 6(a)). Since the expressions are complex-valued, care must be taken in the interpolation: *argument* (i.e., phase) and *modulus* must be independently interpolated to give a new complex value.

Figure 6 shows a comparison of the total wave field as predicted by the UTD and our approximated wave field for the situation shown in Figure 6(a). Although there are differences in the vicinity of the shadow boundary, most properties of the original UTD diffracted field are captured by the approximation: (1) the edge is still the source of the diffracted contribution and path delays are not modified by our approximation, (2) the field is continuous, so no audible artifact is heard when crossing the shadow boundary, (3) the field amplitude is independent of frequency at the shadow boundary, and (4) it decays faster as frequency increases and tends toward the actual value of the UTD diffracted field as the receiver moves away from the shadow boundary. As a result, we conjecture that the spatialized sound produced by our on-line system provides many of the significant cues useful for localization of objects, separation of signals, and comprehension of space in an immersive virtual environment.

5 Simulation Results

The 3D data structures and algorithms described in the preceding sections are implemented in C++ and run both on SGI/Irix and PC/Windows computers. We integrated them into a prototype system that allows a user to move through a virtual environment interactively, while images and spatialized audio are rendered in real-time according to the user's simulated position.

To test if our beam tracing approach is practical for modeling diffraction in typical virtual environments, and to evaluate the benefits of incorporating diffraction into real-time auralization, we ran a series of tests computing propagation paths both with and without diffraction. During each test, we used a 3D model with 1,762 polygons representing one floor of a building (see Figure 8). For simplicity, we assumed that every polygon in the 3D model was 80% reflective and acoustically opaque (no transmission). Before each test, we traced 50,000 beams in breadth-first order from a stationary sound source (located at the white dot in Figure 8) and stored them in a beam tree data structure. Then, as a receiver moved at three inch increments along a hallway (the long vertical one on the right side of each image in Figure 8), we computed propagation paths from source to receiver, updated an impulse response, and auralized spatialized sound in real-time. All the tests were run on a Silicon Graphics Onyx2 workstation using two 195MHz R10000 processors, one of which was dedicated to software convolution of audio signals.

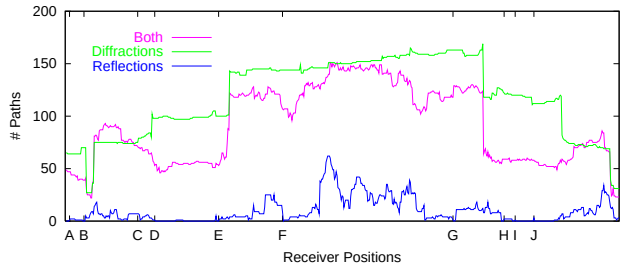
The test sequence was executed three times, once for each of the following beam tracing constraints:

1. **Specular reflection only:** We traced 50,000 beams along paths of specular reflection, with no diffraction. The results represent the state-of-the-art prior to this paper [12, 13].
2. **Diffraction only:** We traced 50,000 beams along paths of diffraction (around silhouette edges into shadow regions), with no specular reflections.
3. **Both specular reflection and diffraction:** We traced 50,000 beams along paths representing arbitrary permutations of specular reflection and diffraction (into shadow regions).

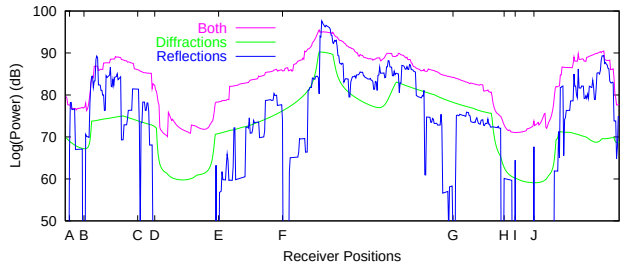
Figure 7 shows plots with the number of propagation paths (the top plot) and the power of the impulse responses (the bottom plot) for each receiver location during the three tests.[1]

From these plots, we confirm that specular reflection alone is not adequate to produce realistic spatialized sound in typical virtual environments. The red curves in Figure 7 show that the number of propagation paths and the power in the corresponding impulse responses varied dramatically with small changes in receiver location. This effect is easily understood by examining images of the beams and power distributions shown in Figure 8(a-d). Note the pattern of thin beams zig-zagging across the hallways in the top-left image. As the receiver walks along the test trajectory, s/he moves in and out of these distinct specular reflection beams, leading to sharp discontinuities in the computed early reverberations. Even worse, there are several locations where no specular reflection paths reached the

[1] Power is computed as $10 \cdot \log \sum_{i=1}^{n} a_i^2$ where n is the number of propagation paths and a_i the amplitude along the ith path.



(a) Number of propagation paths.



(b) Power of computed impulse response in log units.

Figure 7: Plots showing the number of propagation paths (top) and power of the corresponding impulse responses (bottom) computed for every receiver position during tests with specular reflection only (red), diffraction only (blue), and both specular reflection and diffraction (purple). Note how adding diffraction gives a smoothly varying sound level as the receiver moves.

receiver, and thus the power of the auralized sound drops to zero for short periods. These locations, which are marked with capital letters on the horizontal axis of the plots in Figure 7, correspond to the visible "holes" in the beam coverage in Figure 8(a). They are particularly troublesome to users during a walkthrough, as suddenly appearing "dead-zones" clearly fail to match our real-world acoustical experiences.

In contrast, we note that tracing beams only along paths of diffraction leads to smoothly varying reverberations (the blue curves in Figure 7). The reason can be seen in Figure 8(e-h). Diffraction beams tend to cover larger volumes of space than specular beams; and, in our test, they collectively cover all reachable parts of the 3D environment (the light gray regions in the middle of the model correspond to elevator and wiring shafts unreachable by sound). Even though we traced diffraction beams only into shadow regions, our approximation produced a smoothly varying impulse response (see the power map in Figure 8(f)) because direct paths were replaced by diffracting ones with equal amplitude at shadow boundaries as the receiver moved past open doors.

Finally, the test with both specular reflections and diffractions (the purple curves in Figure 7) shows that the power varies quite smoothly, while early reflections and diffractions due to the environment are clearly evident. The improvement in reverberation can be seen clearly by examining the echograms (temporal plot of scattered energy reaching the receiver) shown in the rightmost images of Figure 8. Each pulse in these plots corresponds to a propagation path (shown in yellow in the third column of Figure 8). Note that the echogram measured with both specular reflections and diffractions contains not only the shortest (diffracted) path from the source (the left-most spike), but it also has many high-power early contributions not found in the other tests because they are reflections of previously diffracted waves, or diffractions of previously reflected waves. These contributions combine with the earliest arriving sound wave to provide the dominant acoustical cues.

We also gathered computational statistics during the three tests (Table 1). Column 2 shows the rate (in beams/second) at which beams are traced from the stationary source location in each test. Column 3 shows the rate (in paths/second) at which propagation sequences are computed and processed to form propagation paths to the moving receiver location, including calculation of reflection and diffraction coefficients. The next three columns show the average number of transmissions through transparent cell boundaries (Trans), specular reflections (Refl), and diffractions (Diff) along the computed paths. Finally, the right-most column (Update Time(s)) shows the time (in seconds) required to update the impulse response for each new receiver location in the three tests. Based on these results, we conclude that tracing propagation paths with both diffractions and specular reflections is quite practical for interactive virtual environment applications. Although diffraction increases the time required to trace beams and construct propagation paths (by almost $2\times$), the system still updates impulse responses at interactive time steps (every 49ms) with our method.

Test Name	Compute Rates		Path Statistics			Update Time (s)
	Beams/s	Paths/s	Trans	Refl	Diff	
Specular	6,305	4,289	4.8	3.9	0.0	0.002
Diffract	3,173	1,190	6.8	0.0	5.2	0.163
Both	3,778	2,943	4.2	1.8	1.7	0.049

Table 1: Beam tracing and path generation statistics.

Figure 9 shows visualizations of our results for different application domains. The left-most pair of images shows the power of sound reaching different parts of a city from a siren located on top of a building. In this case, diffraction due to edges of large buildings is the dominant acoustical effect. The second set of images explores the acoustical variations of different seats in the Opéra de la Bastille theater in Paris. There, diffraction over the lip of the orchestra pit provides the primary means for sound to reach the audience, and the slanted balconies are responsible for significant occlusion and diffraction effects. Finally, the rightmost pair of images shows how spatialized sound with diffraction can be used to enhance an interactive video game, as sound diffracting through non-axial obstacles and over walls helps players find each other.

6 Discussion

Our current beam tracing implementation is practical only for densely-occluded and coarsely detailed 3D models, since beams get fragmented in scenes with many free-space cell boundaries [12, 39]. However, since this class of models contains many types of interesting acoustical environments (e.g., buildings and cities), our system is useful for the proposed application domain. To work well for sound simulations in detailed 3D models, our beam tracing algorithm would have to be enhanced, possibly by extending Fortune's topological beam tracing method [11] to work for edge sources.

The UTD is an approximate model of sound propagation, valid mostly for high frequencies and infinite wedges. However, our technique for enumerating propagation sequences can be used in combination with other theories, such as [38], in the context of off-line simulations. The accuracy would be improved, especially at low frequencies and near-field from the wedges. Also, as mentioned in [41], the shortest paths constructed by our technique would still be useful for efficient auralization even if the diffraction coefficients (or filters) are derived from another theory.

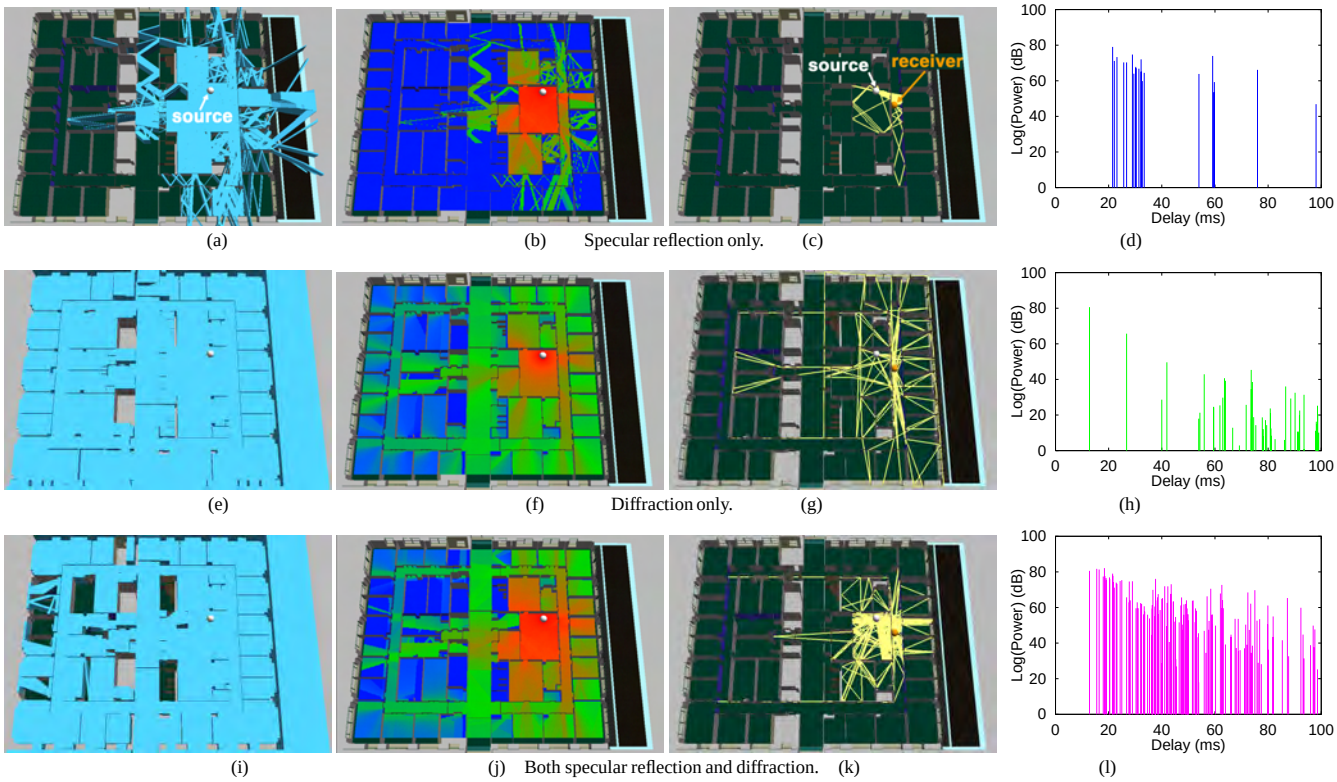


Figure 8: Images depicting results of our experiment with 50,000 beams for specular reflection only (top row), diffraction only (middle row), and both specular reflection and diffraction (bottom row). The left-most column shows beams (cyan) traced from a source point. The next column shows the corresponding power plots (red represents higher power and green lower power). The third column shows computed propagation paths to receiver point. Finally, the right-most column shows the corresponding echograms. Note how diffraction beams fill the entire space and how diffraction combines with reflection to produce a more complete acoustical impression.

7 Conclusion and Future Work

In this paper we introduce an efficient technique for incorporating diffraction effects in interactive audio simulations for virtual environments. Relying upon the Uniform Theory of Diffraction, we describe a beam-tracing approach to construct propagation paths with diffraction, and we introduce a practical approximation to the diffracted field in shadow regions. This is the first instance where a realistic, physically-based, diffraction model is used to produce sound at interactive rates in complex virtual environments. By simulating diffraction, we remove the disturbing "cuts" in the audio that occur when a sound source is occluded by an acoustically opaque surface, and make it possible to localize occluded sound sources. Based on our initial experiences, we conclude that it is possible to compute diffraction paths in real-time and that diffraction dramatically improves the realism and quality of the audio experience.

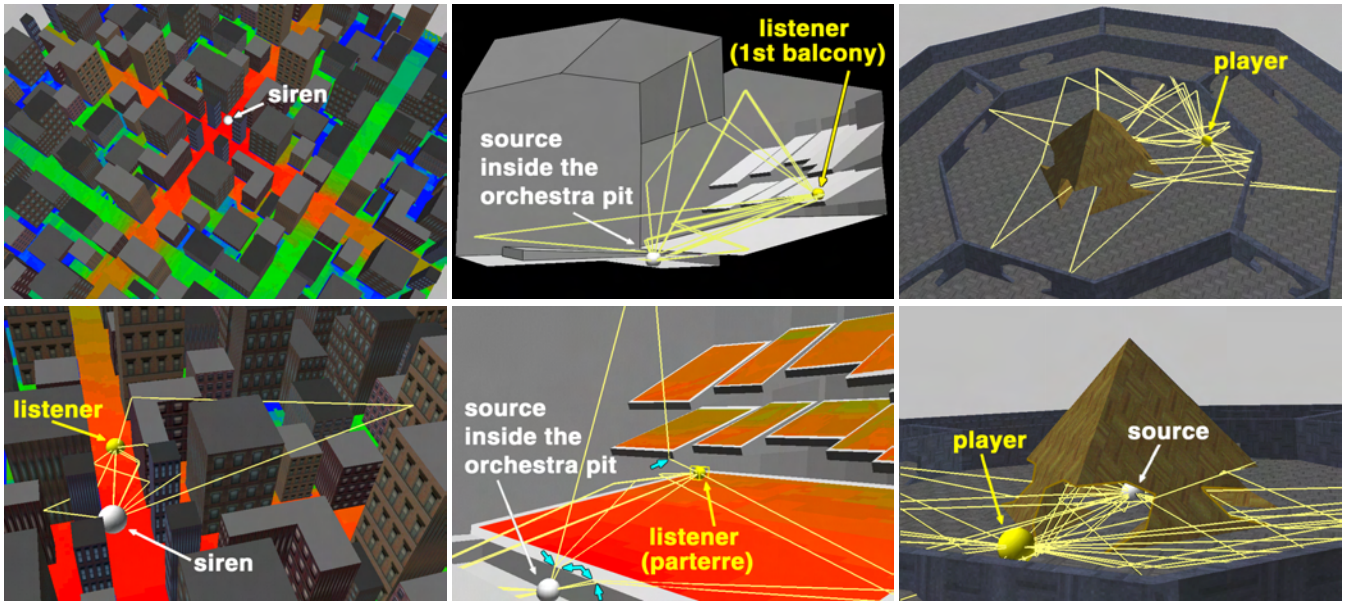
This research suggests several directions for further study. First, evaluation of simulation results by comparison to measured data is essential. Towards this end, we have recently built a "room" whose walls have known acoustical bidirectional reflectance distribution functions and that allows addition of panels to create interesting geometries, including diffracting panels. We are using this room to verify our simulations and evaluate different approximations. Perceptual assessments, which are probably the most important for virtual environments, will also be conducted.

Second, application of the proposed methods to problems beyond acoustics is a promising topic for future work. We are currently investigating hybrid beam tracing and path tracing approaches to global illumination in which coarsely detailed beams are used to guide the sampling and intersection of paths in a Monte Carlo lighting simulation. This would also be useful to efficiently simulate diffuse surface reflection for acoustic simulations [7, 42]. Other potential applications include motion planning, transmitter power prediction, fire simulation, and traffic analysis.

Finally, perhaps the most interesting topic for future work is the study of the inter-play between visual and auditory stimuli in human perception of 3D environments. Accurate simulations of both sound and light in an interactive system may provide a useful tool for perceptual psychologists to investigate this important question.

Acknowledgments

The authors would like to thank Sid Ahuja for supporting this research. Steve Fortune and Roland Freund advised us on the construction of the diffraction paths. Gary Elko, Mohan Sondhi and Jim West also provided valuable advise about sound diffraction and Agata Opalach helped design the figures in this paper. Many thanks to Fredo Durand and George Drettakis for their feedback on an early version of the paper and to the anonymous reviewers for their helpful comments.



(a) City (1125 polygons) (b) Opéra de la Bastille (117 polygons) (c) Arena (665 polygons).

Figure 9: Visualizations of sound simulations for different applications. (a) Acoustic power coverage map for a siren in a city environment. (b) Study of early propagation paths for a source located in the orchestra pit of an opera house. Note the diffracted paths over the lip of the pit and balconies (cyan arrows). (c) Diffracted sound paths allow players of a video game to localize the source hidden under the pyramid.

References

[1] J.M. Airey, J.H. Rohlf, and F.P. Brooks, Jr. Towards image realism with interactive update rates in complex virtual building environments. In Rich Riesenfeld and Carlo Séquin, editors, *Computer Graphics (1990 Symposium on Interactive 3D Graphics)*, pages 41–50, March 1990.

[2] L. Aveneau and M. Meriaux. Rendering polygonal scenes with diffraction account. *Seventh International Conference in Central Europe on Computer Graphics and Visualization (Winter School on Computer Graphics)*, February 1999.

[3] L. Aveneau, Y. Pousset, R. Vauzelle, and M. Mériaux. Development and evaluations of physical and computer optimizations for the 3d utd model. *AP2000 Millennium Conference on Antennas & Propagation (poster)*, April 2000.

[4] H.L. Bertoni. Coverage prediction for mobile radio systems operating in the 800/900 MHz frequency range. *IEEE Transactions on Vehicular Technology (Special Issue on Mobile Radio Propagation)*, 37(1), February 1988.

[5] J. Blauert. *Spatial Hearing : The Psychophysics of Human Sound Localization*. M.I.T. Press, Cambridge, MA, 1983.

[6] M. Born and E. Wolf. *Principles of Optics*. 7th ed., Pergamon Press, 1999.

[7] B.-I. L. Dalenbäck. Room acoustic prediction based on a unified treatment of diffuse and specular reflection. *J. of the Acoustical Soc. of America*, 100:899–909, 1996.

[8] G. Drettakis. *Structured Sampling and Reconstruction of Illumination for Image Synthesis*. PhD thesis, University of Toronto, January 1994.

[9] N.I. Durlach and A.S. Mavor. Virtual reality scientific and technological challenges. National Research Council Report, National Academy Press, 1995.

[10] P. Filippi, D. Habault, J.P. Lefevre, and A. Bergassoli. *Acoustics, basic physics, theory and methods*. Academic Press, 1999.

[11] S.J. Fortune. Topological beam tracing. In *Proc. 15th ACM Symposium on Computational Geometry*, pages 59–68, 1999.

[12] T. Funkhouser, I. Carlbom, G. Elko, G. Pingali, M. Sondhi, and J. West. A beam tracing approach to acoustic modeling for interactive virtual environments. *ACM Computer Graphics, Proc. SIGGRAPH98*, pages 21–32, July 1998.

[13] T. Funkhouser, P. Min, and I. Carlbom. Real-time acoustic modeling for distributed virtual environments. *ACM Computer Graphics, Proc. SIGGRAPH99*, pages 365–374, August 1999.

[14] J. Goodman and J. O'Rourke, editors. *Handbook of Discrete and Computational Geometry*. CRC Press, 1997.

[15] P. Heckbert and P. Hanrahan. Beam tracing polygonal objects. *Computer Graphics (SIGGRAPH84)*, 18(3):119–127, July 1984.

[16] D.C. Hothersall, S.N. Chandler-Wilde, and M.N. Hajmirzae. Efficiency of single noise barriers. *J. of Sound and Vibration*, 146(2):303–322, 1991.

[17] C. Huygens. *Traité de la Lumiere*. London, Macmillan & Co., 1912.

[18] P. Jean. A variational approach for the study of outdoor sound propagation and application to railway noise. *J. of Sound and Vibration*, 212(2):275–294, 1998.

[19] C. B. Jones. A new approach to the `hidden line' problem. *Computer Journal*, 14(3):232–237, August 1971.

[20] T. Kawai. Sound diffraction by a many sided barrier or pillar. *J. of Sound and Vibration*, 79(2):229–242, 1981.

[21] J.B. Keller. Geometrical theory of diffraction. *J. of the Optical Society of America*, 52(2):116–130, 1962.

[22] S.C. Kim, B. Guarino, T. Willis, V. Erceg, S. Fortune, R. Valenzuela, L. Thomas, J. Ling, and J. Moore. Radio propagation measurements and prediction using three-dimensional ray tracing in urban environments at 908 MHz and 1.9 GHz. *IEEE Trans. on Vehicular Technology*, 48:931–946, 1999.

[23] M. Kleiner, B.I. Dalenbäk, and P. Svensson. Auralization - an overview. *J. of the Audio Engineering Society*, 41(11):861–875, November 1993.

[24] R.G. Kouyoumjian and P.H. Pathak. A uniform geometrical theory of diffraction for an edge in a perfectly conducting surface. *Proc. of IEEE*, 62:1448–1461, November 1974.

[25] H. Lehnert and J. Blauert. Principles of binaural room simulation. *Applied Acoustics*, 36:259–291, 1992.

[26] D.A. McNamara, C.W.I. Pistorius, and J.A.G. Malherbe. *Introduction to the Uniform Geometrical Theory of Diffraction*. Artech House, 1990.

[27] H. Medwin, E. Childs, and G. Jebsen. Impulse studies of double diffraction: A discrete huygens interpretation. *J. Acoust. Soc. Am.*, 72:1005–1013, 1982.

[28] P. Min and T. Funkhouser. Priority-driven acoustic modeling for virtual environments. *Proc. Eurographics'2000*, 2000.

[29] J. S. B. Mitchell. Geometric shortest paths and network optimization. In Jörg-Rüdiger Sack and Jorge Urrutia, editors, *Handbook of Computational Geometry*. Elsevier Science Publishers B.V. North-Holland, Amsterdam, 1998.

[30] B. C.J. Moore. *An introduction to the psychology of hearing*. Academic Press, 4th ed., 1997.

[31] A.D. Pierce. *Acoustics. An introduction to its physical principles and applications*. 3rd ed., American Institute of Physics, 1984.

[32] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C, 2nd ed*. Cambridge University Press, New York, 1992.

[33] A. Rajkumar, B.F. Naylor, F. Feisullin, and L. Rogers. Predicting RF coverage in large environments using ray-beam tracing and partitioning tree represented geometry. *Wireless Networks*, 2(2):143–154, 1996.

[34] L. Savioja, J. Huopaniemi, T. Lokki, and R. Väänänen. Creating interactive virtual acoustic environments. *J. of the Audio Engineering Society*, 47(9):675–705, September 1999.

[35] J. Stam. Diffraction shaders. *ACM Computer Graphics, Proc. SIGGRAPH99*, pages 101–110, August 1999.

[36] U. Stephenson and U. Kristiansen. Pyramidal beam tracing and time dependent radiosity. *15th International Congress on Acoustics*, pages 657–660, June 1995.

[37] R. L. Storms. *Auditory-Visual Cross-Modal Perception Phenomena*. PhD thesis, Naval Postgraduate School, Monterey, California, September 1998.

[38] U. P. Svensson, R. I. Fred and J. Vanderkooy Analytic secondary source model of edge diffraction impulse responses. *J. of the Acoustical Society of America*, 106:2331–2344, 1999.

[39] S. Teller. Computing the antiumbra cast by an area light source. *Computer Graphics (SIGGRAPH92)*, 26(2):139–148, 1992.

[40] S. Teller. *Visibility Computations in Densely Occuded Polyhedral Environments*. PhD thesis, Computer Science Div., Univ. of California, Berkeley, 1992.

[41] R. Torres, P. Svensson and M. Kleiner. Computation of edge diffraction for more accurate room acoustics auralization. *J. of the Acoustical Society of America*, 109:600–610, 2001.

[42] R. Torres. Studies of Edge Diffraction and Scattering: Applications to Room acoustics and Auralization. PhD thesis, Dept. of Applied Acoustics, Chalmers University of Technology, Sweden, 2000.

[43] N. Tsingos and J.-D. Gascuel. Soundtracks for computer animation: sound rendering in dynamic environments with occlusions. *Proceedings of Graphics Interface'97*, pages 9–16, May 1997.

A Auralizing the wedge diffracted field

According to the UTD, the acoustic pressure field diffracted by a wedge can be expressed in terms of the incident field on the edge, $\mathcal{E}_{\text{incident}}(M)$, as:

$$\mathcal{E}_{\text{diffracted}}(R) \quad = \mathcal{E}_{\text{incident}}(M)\ D\ A(r,\rho)\ e^{-ikr}, \tag{3}$$

where R is the receiver location, M is the diffraction point (see Figure 2) , $A(r,\rho) = \sqrt{\rho r/(\rho + r)}$ is a scalar distance attenuation term along the propagation path, the complex exponential e^{-ikr} represents phase variation along the diffracted path, $k = 2\pi/\lambda$ is the wave number (λ is the wavelength). Equation (3) is applied successively for every diffracting wedge and multiplied by attenuations due to reflections, transmissions, and path length to form a contribution to the impulse response for every propagation path.

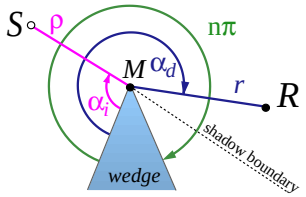


Figure 10: Notations for the UTD diffraction coefficient.

D is the complex-valued UTD diffraction coefficient [24, 26] accounting for amplitude and phase changes due to diffraction:

$$\begin{aligned} D(n,k,\rho,r,\theta_i,\alpha_i,\alpha_d) = & -\frac{e^{-i\frac{\pi}{4}}}{2n\sqrt{2k\pi}\sin\theta_i} \\ & \Big[\tan^{-1}\left(\tfrac{\pi+(\alpha_d-\alpha_i)}{2n}\right) F\left(kLa^+(\alpha_d-\alpha_i)\right) \\ & + \tan^{-1}\left(\tfrac{\pi-(\alpha_d-\alpha_i)}{2n}\right) F\left(kLa^-(\alpha_d-\alpha_i)\right) \\ & + \Big\{ \tan^{-1}\left(\tfrac{\pi+(\alpha_d+\alpha_i)}{2n}\right) F\left(kLa^+(\alpha_d+\alpha_i)\right) \\ & + \tan^{-1}\left(\tfrac{\pi-(\alpha_d+\alpha_i)}{2n}\right) F\left(kLa^-(\alpha_d+\alpha_i)\right) \Big\} \Big], \end{aligned} \tag{4}$$

where (see also Figure 10 and Figure 2):

$$F(X) = 2i\sqrt{X}e^{iX}\int_{\sqrt{X}}^{+\infty} e^{-i\tau^2}d\tau, \tag{5}$$

$$L = \frac{\rho r}{\rho + r}\sin^2\theta_i, \tag{6}$$

$$a^{\pm}(\beta) = 2\cos^2\left(\frac{2\pi n N^{\pm} - \beta}{2}\right), \tag{7}$$

$N^{\pm}$ is the integer that satisfies more closely the relations:

$$2\pi n N^+ - \beta = \pi \quad \text{and} \quad 2\pi n N^- - \beta = -\pi \tag{8}$$

Several approximations exist in the related literature, useful for implementation of Eq. 4. In particular, relations (8) reduce to:

$$N^+ = \begin{cases} 0 & \text{for } \beta \le \pi(n-1) \\ 1 & \text{for } \beta > \pi(n-1) \end{cases}, \qquad N^- = \begin{cases} -1 & \text{for } \beta < \pi(1-n) \\ 0 & \text{for } \pi(1-n) \le \beta \le \pi(1+n) \\ 1 & \text{for } \beta > \pi(1+n) \end{cases}, \tag{9}$$

and Kawai [20] gives an approximate rational expression for the integral in Eq. (5):

$$\begin{aligned} &\text{for } X < 0.8 : F(X) = \sqrt{\pi X}\left(1 - \frac{\sqrt{X}}{0.7\sqrt{X}+1.2}\right) e^{i\frac{\pi}{4}}\sqrt{\frac{X}{X+1.4}} \\ &\text{for } X \ge 0.8 : F(X) = \left(1 - \frac{0.8}{(X+1.25)^2}\right) e^{i\frac{\pi}{4}}\sqrt{\frac{X}{X+1.4}} \end{aligned} \tag{10}$$

Cotangent terms in Equation (4) are still singular at a reflection or shadow boundary and cannot be evaluated numerically at these boundaries. However, in the vicinity of such a boundary we can express the terms $\alpha_i \pm \alpha_d$ as $\beta = 2\pi n N^{\pm} \mp (\pi - \varepsilon)$. The coefficient is continuous and its value can be computed using [24]:

$$\tan^{-1}\left(\tfrac{\pi\pm\beta}{2n}\right) F\left(kLa^{\pm}(\beta)\right) \simeq n e^{-i\pi/4}\left(\sqrt{2\pi kL}\,\mathrm{sgn}(\varepsilon) - 2kL\varepsilon e^{-i\pi/4}\right),$$

where $\mathrm{sgn}(\varepsilon) = 1$ if $\varepsilon > 0$ and -1 otherwise.

In order to render the virtual sound field, we compute a digital filter [23, 25], with which audio signals emanating from the source can be convolved to produce a spatialized audio signal with reverberation. For high quality auralization, this filter is computed using complex values in Fourier frequency space at the desired sampling rate resolution. For interactive applications, fewer frequency bands can be considered, depending on how much processing power is available. The modulus of the complex field for the center frequency of each frequency band can be used to *re-equalize* the source signal. For more information on the signal processing involved in auralization, please refer to [23, 25, 34]

Validating Acoustical Simulations in the Bell Labs Box

Nicolas Tsingos
INRIA, France

Ingrid Carlbom
Bell Laboratories

Gary Elko and Robert Kubli
MH Acoustics

Thomas Funkhouser
Princeton University

We developed the Bell Labs Box to compare the result of acoustical simulations with measured data in simple and controlled settings.

Computer simulated sound propagation through 3D environments is important in many applications, including computer-aided design, training, and virtual reality. In many cases, the accuracy of the acoustical simulation is critical to an application's success. For example, in concert hall and factory design (where designers must meet US Occupational Safety and Health Administration [OSHA] sound limits), a simulation's accuracy might save costly reengineering after construction. In virtual environments, experiments have shown that more accurate acoustic modeling provides a stronger sense of presence.[1] Furthermore, auditory cues help form spatial impressions, separate simultaneous sound signals, and localize objects,[2] such as when a soldier locates an enemy in a training exercise or a firefighter locates a person stranded in a burning building. In contrast, incorrect auditory cues can lead to negative training.[3,4]

Although several systems (such as the Bose Modeler [http://www.bose.com], CATT-Acoustic [http://www.netg.se/catt], and Odeon [http://www.dat.dtu.dk/~odeon/]) are available for computing sound propagation in 3D environments, there hasn't been a detailed evaluation of their accuracy. The primary reason is that the acoustics of most real-world environments is complex, and thus detailed quantitative comparison of computed impulse responses is difficult. As a result, acousticians have resorted to comparing gross statistics of impulse responses (for example, reverberation time and clarity) and/or using human listening tests to validate computer simulations.[5,6]

This is similar to the situation in computer graphics during the mid-1980s. At that time, there were several global illumination algorithms that simulated the propagation of light through a 3D environment and produced an image for a given camera—for example, ray tracing[7] and radiosity.[8] Yet, there were few results regarding these simulations' accuracy. Computer-generated pictures and corresponding photographs rarely matched, and little insight into the causes of the mismatch could be derived from the comparisons due to the complexity of light transport in most scenes.

In response to this situation, researchers at Cornell University[9] constructed a simple real-world scene, the Cornell Box, in which they carefully measured the lighting, geometry, and reflectance properties of every surface and duplicated it in a 3D computer graphics model. Using this simple and controlled experimental setup, they were able to make meaningful comparisons between global illumination simulations and photographs of the scene and provide explanations for the differences. (See the sidebar "Background and Previous Work" for more details on earlier work.)

Motivated by the Cornell Box's success, we've built a simple experimental setup for validating sound propagation simulations. Our setup, the Bell Labs Box, comprises a simple configuration of planar surfaces with a speaker and a microphone (see Figure 1 on p. 30). Although the room's basic configuration is a simple six-sided box, we constructed it with reconfigurable panels that we can insert or remove to create various interesting geometries, including ones with diffracting panels.

The key ideas behind this experimental setup are simplicity and control. First, the room has just a few planar surfaces, so we can easily compute possible sequences of reflections and diffractions in a simulation and recognize them in an impulse response. Second, we can independently measure the speaker radiation pattern, microphone directivity pattern, and reflectance properties of every surface in an anechoic chamber, providing simulation parameters that closely match the real-world environment. Finally, the room is configurable, providing a mechanism by which we can validate acoustical simulations with various geometric effects. These features let us compare the results of simulations with measurement data in a simple and controlled setting. Our goal is to obtain a detailed understanding of the limitations of the simulations in these simple environments, and then apply the acquired knowledge to improve our simulations with increased model complexity.

0272-1716/02/$17.00 © 2002 IEEE

Background and Previous Work

The problem we're addressing in this article is the validation of a solution to an integral equation expressing the wavefield (acoustic pressure and particle velocity) at some point in space in terms of the wavefield at other points (or equivalently on surrounding surfaces). For light simulations, Kajiya's rendering equation[1] describes the wave equation. For sound simulations, it is described by the Helmoltz–Kirchoff integral theorem,[2] which incorporates time and phase dependencies.

Because sound and light are both wave phenomena, the methods for their simulation and validation share many features. Yet, sound has different characteristics from light that introduce new and interesting problems:

- *Wavelength*. The wavelengths of audible sound are five to seven orders of magnitude longer than visible light, ranging between 0.02 and 17 meters (for 20 KHz and 20 Hz, respectively). Diffraction of sound occurs around obstacles of the same size as the wavelength (such as tables) and reflections are primarily specular for large, flat surfaces (such as walls). Small objects (like coffee mugs) have little effect on the sound field (for all but the highest wavelengths). As a result, when compared to computer graphics, acoustics simulations tend to use 3D models with far less geometric detail. However, they must find propagation paths with specular reflections and diffractions efficiently, and they must consider the effects of different obstacles at a range of wavelengths.
- *Speed*. At 343 meters per second, the speed of sound in air is six orders of magnitude less than light, and sound propagation delays are perceptible to humans. Thus, acoustic models must compute the propagation paths' exact time/frequency distribution, and the source sound must be auralized by convolution with the corresponding impulse response. This digital filter represents the delay and amplitude of the sound arriving along different propagation paths. In contrast, we can ignore the propagation delay of light and must only compute the energy steady-state response.
- *Coherence*. Sound is a coherent wave phenomenon, and interference between out-of-phase waves can be significant. Accordingly, acoustical simulations must consider phase when summing the cumulative contribution of many propagation paths to a receiver. More specifically, because the phase of the wave traveling along each propagation path is determined by the path length, acoustical models must compute accurate path lengths (up to a small percentage of the wavelength). In contrast, most light sources (except lasers) emit largely incoherent waves, and thus lighting simulations simply sum the power of different propagation paths.
- *Dynamic range*. The human ear is sensitive to five-orders-of-magnitude difference in sound amplitude, and arrival time differences make some high-order reflections audible.[3,4] Therefore, compared to computer graphics, acoustical simulations usually aim to compute several times more reflections, and the statistical time–frequency effects of late sound reverberation are more significant than for global illumination.

Despite these differences, the problems of simulation and validation are similar for both sound and light. For simulation, the main difficulty arises from the wavefield discontinuities caused by occlusions, caustics, and specular highlights, resulting in large variations over small portions of the integration domain (that is, surfaces and/or directions). Due to these discontinuities, no general-purpose, analytic formula can describe the wavefield at a given point, and solutions must rely on sampling or subdivision of the integration domain into components that we can solve efficiently and accurately. Traditionally, four approaches have been used to address this problem: finite- or boundary-element methods,[5–7] recursive ray tracing,[8,9] Monte Carlo path tracing,[1,10] and beam tracing.[11–13] All four methods have been used for sound and light.

For validation, the common problems are defining a quantitative measure of accuracy and understanding the causes of simulation errors. We generally address the first problem in acoustics with statistical measures of impulse responses and with human listening tests. The second problem has largely been explained by conjecture. For example, recent round-robin studies applied several computer simulation tools to the same concert hall model and compared their output to in-situ measurements.[14,15] Although these studies provide a nice reference for comparing simulations, they differ from our work in several respects. First, they focus on complicated concert halls rather than simple rooms like ours. Thus, it's difficult to evaluate the correctness of any of the simulations—that is, the acoustics are so complex that none of the tools get the right answer. Second, they compare only gross statistical measures of impulse responses[16] (for example, T30, EDT, D50, C80, TS, G, LF, LFC, IACC, and ISO 3328), so it's difficult to determine the causes for simulation errors. Is it because we ignored propagation due to edge diffraction or because we didn't model angle-dependent reflection functions correctly? Without detailed examination of simple impulse responses, these questions are difficult to answer.

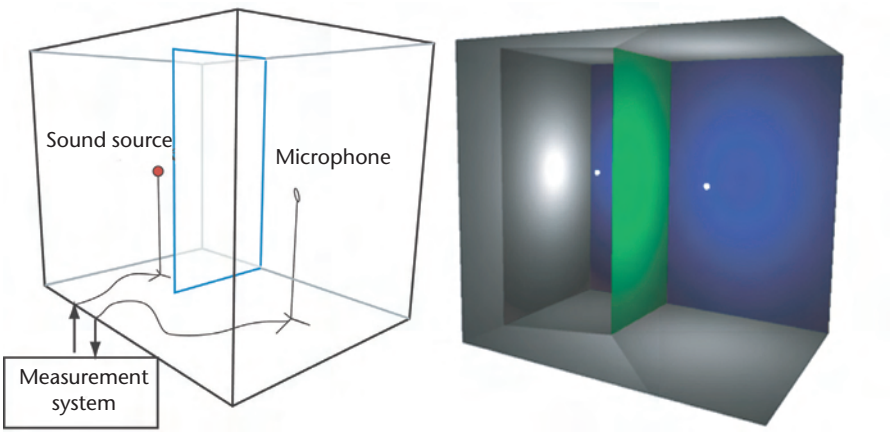
Our work differs from most previous validation studies in that we focus only on simple and controlled acoustical environments. In contrast to previous simulation systems, we not only use a simple 3D model of the environment, but the environment itself is simple—a box with a few configurable panels. In this respect, our work resembles recent detailed comparisons of geometrical simulations and measurements.[17,18] These studies show that geometrical room acoustics can lead to satisfying simulations. Yet, Suh and Nelson[17] don't take into account sound diffraction, and Torres et al.[18] don't use reverberant environments. In contrast, we present a simulation system and measurement environment that treats the combined effects of specular reflection and diffraction in a reverberant environment. As a result, we're generally able to simulate the sound propagation quite accurately, and we can evaluate computed impulse responses with great accuracy.

continued on p. 30

continued from p. 29

References

1. J.T. Kajiya, "The Rendering Equation," *Computer Graphics* (Proc. Siggraph 86), vol. 20, no. 4, 1986, pp. 143-150.
2. M. Born and E. Wolf, *Principles of Optics*. 7th ed., Cambridge Univ. Press, Cambridge, UK, 1999.
3. E. Zwicker and H. Fastl, *Psychoacoustics: Facts and Models*, Springer, New York, 1999.
4. J. Blauert, *Spatial Hearing: The Psychophysics of Human Sound Localization*, MIT Press, Cambridge, Mass., 1997.
5. C.M. Goral et al., "Modeling the Interaction of Light Between Diffuse Surfaces," *Computer Graphics* (Proc. Siggraph 84), vol. 18, no. 3, July 1984, pp. 213-222.
6. G.R. Moore, *An Approach to the Analysis of Sound in Auditoria*, doctoral thesis, Cambridge Univ., UK, 1984.
7. R.D. Ciskowski and C.A. Brebbia, eds., *Boundary Element Methods in Acoustics*, Elseiver Applied Science, London, 1991.
8. U.R. Krockstadt, "Calculating the Acoustical Room Response by the Use of a Ray Tracing Technique," *J. Sound and Vibrations*, vol. 8, no. 18, 1968.
9. T. Whitted, "An Improved Illumination Model for Shaded Display," *C. ACM*, vol. 23, no. 6, June 1980, pp. 343-349.
10. J.M. Naylor, "Odeon - Another Hybrid Room Acoustical Model," *Applied Acoustics*, vol. 38, no. 1, 1993, pp. 131-143.
11. J. Martin, D. van Maercke, and J.P. Vian, "Binaural Simulation of Concert Halls: A New Approach for the Binaural Reverberation Process," *J. Acoustical Soc. of America*, vol. 94, no. 6, Dec. 1993, pp. 3255-3263.
12. P. Heckbert and P. Hanrahan, "Beam Tracing Polygonal Objects," *Computer Graphics* (Proc. Siggraph 84), vol. 18, no. 3, July 1984, pp. 119-127.
13. T. Funkhouser et al., "A Beam Tracing Approach to Acoustic Modeling for Interactive Virtual Environments," *Computer Graphics* (Proc. Siggraph 98), July 1998, pp. 21-32.
14. M. Vorlander, "International Round Robin on Room Acoustical Computer Simulations," *Proc. 15th Int'l Congress of Acoustics*, 1995.
15. I. Bork, "A Comparison of Room Simulation Software - The 2nd Round Robin on Room Acoustical Computer Simulation," *Acoustica*, vol. 86, no. 6, 2000, pp. 943-956.
16. L.L. Beranek, *Concert and Opera Halls: How They Sound*, American Inst. of Physics, New York, 1996.
17. J.S. Suh and P.A. Nelson, "Measurment of Transcient Responses of Rooms and Comparison with Geometrical Acoustic Models," *J. Acoustical Soc. of America*, vol. 105, no. 4, Apr. 1999, pp. 2304-2317.
18. R. Torres, P. Svensson, and M. Kleiner, "Computation of Edge Diffraction for More Accurate Room Acoustics Auralization," *J. Acoustical Soc. of America*, vol. 109, 2001, pp. 600-610.



(a) **(b)**

1 (a) A schematic view of the Bell Labs Box with measurement apparatus. (b) A computer graphics rendering of the Bell Labs Box lit by two point light sources.

Simulation system

Our sound simulation is based on the beam-tracing method described in Funkhouser et al.[10] and Tsingos et al.[11] One major advantage of this method over alternative approaches is that it finds all propagation paths with arbitrary combinations of transmission, specular reflection, and diffraction without spatial aliasing. This feature is particularly important for our studies with diffracting panels (see Figure 2) inside the Bell Labs Box.

The system takes as input

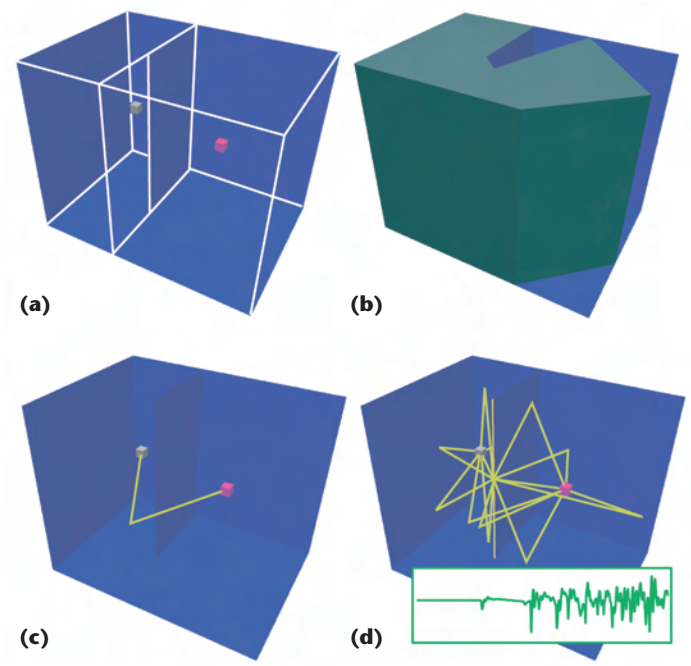
- a 3D environment described as a set of polygons with frequency-dependent impedances or bidirectional scattering filters,
- a speaker described by its location and angular radiation pattern, and
- a microphone described by its location and optional angular directivity pattern.

The system outputs a simulated impulse response.

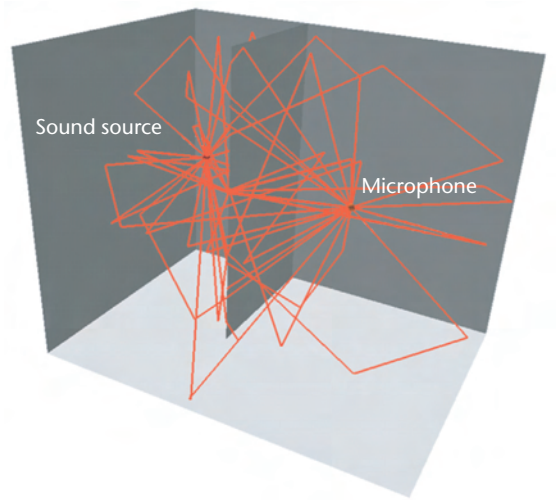
The system executes in four steps, as Figure 3 shows. In the first step, we build a spatial subdivision data structure representing a binary space partition of 3D space into convex polyhedral cells. The purpose of this step is to decompose space into cells whose boundaries are aligned with polygons of the 3D input model and whose adjacencies are stored explicitly in a graph structure to enable efficient traversals of 3D space during beam tracing.

In the second step, we trace the convex polyhedral beams representing different propagation sequences through cells of the spatial subdivision. The traversal starts in the cell containing the speaker location with a beam representing the entire cell. Next, it visits adjacent cells iteratively, considering different permutations of transmissions, specular reflections, and diffractions due

2 We can mount additional panels inside the Bell Labs Box to study the effects of sound diffraction.



3 The four phases of our simulation system are (a) spatial subdivision, (b) polyhedral beam tracing, (c) path construction, and (d) impulse-response generation.



4 All 45 possible propagation paths combining two specular reflections and one diffraction around the edge of the panel in a model of the Bell Labs Box.

to the faces and edges on the current cell's boundary. As the algorithm traverses a cell boundary into a new cell, a copy of the current convex pyramidal beam is clipped by the boundary to include the region of space passing through the convex polygonal boundary to model transmissions. At each reflecting cell boundary, a copy of the transmission beam is mirrored across the plane supporting the cell boundary to model specular reflections. At each diffracting wedge, a new beam is spawned whose source is the edge and whose extent includes all rays predicted by the Geometrical Theory of Diffraction.[12] The traversal along any sequence terminates when either the length of the shortest path within the beam or the cumulative attenuation exceed some user-specified thresholds. The traversal may also terminate when the total number of beams traced or the elapsed time exceed other thresholds.

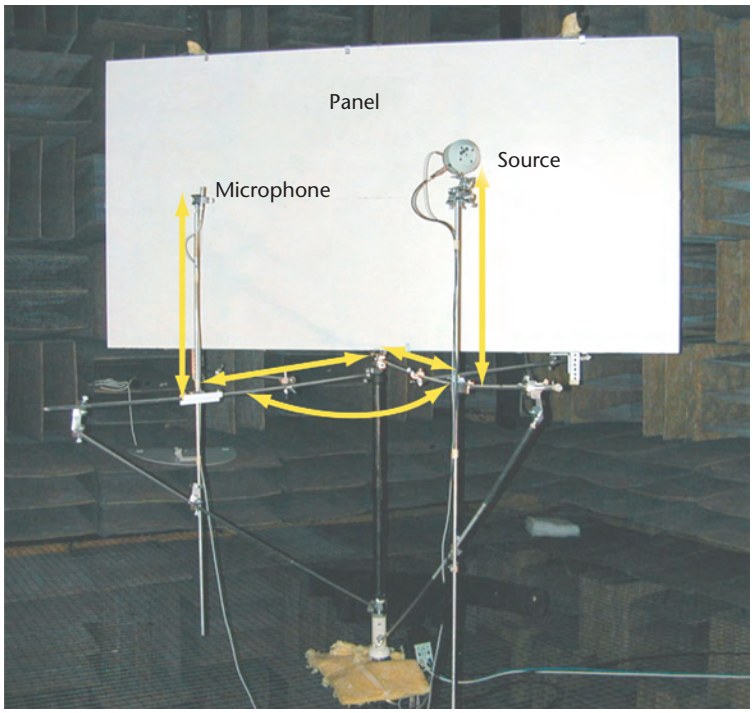
In the third step, for each beam containing the microphone location, we compute the shortest propagation path from the speaker to the microphone along the sequence of transmissions, diffractions, and specular reflections represented by the beam (see Figure 4). We uniquely determine from the locations the intersections with specularly reflecting faces by the locations of the speaker, microphone, and the intersections with diffracting edges (diffraction points). We find the diffraction points by solving a nonlinear system of equations expressing equal angle constraints at diffracting edges. Once the diffraction points are found, we construct a piecewise-linear polyline representing the path along which sound travels from source to receiver along the propagation sequence. From this path, we compute a length-, angle-, and frequency-dependent filter.

In the final step, for each valid propagation path from the speaker to the microphone, we add its contribution to the simulated impulse response. Our implementation includes source and material filtering effects derived from either measurements (see the next section for details) or analytical models. We compute diffraction coefficients using the Uniform Theory of Diffraction[13] (see Tsingos et al.[11] for details). We also take into account atmospheric scattering following the ISO 9013-1 specifications. All calculations are performed in a complex Fourier domain at the sampling rate resolution. Our current system uses a sampling rate of 51,200 Hz—the sampling rate of our source and material measurements. Thus, for a 1-second response, we're computing 25,600 complex coefficients per path. The simulated transfer function is the sum of the coefficients for all paths, and the final impulse response is the inverse Fourier transform of the sum.

After all beams have been traced up to a user-specified termination criterion and the contribution of all propagation paths have been summed, we output the resulting impulse response for comparison to measurements.

5 (a) Outside view of the Bell Labs Box. (b) Inside view of the six-sided Bell Labs Box (without diffraction panel).

6 Our reference source is the Brüel & Kjaer artificial mouth.



7 Measurement setup for bidirectional material properties in Bell Labs' anechoic chamber. In particular, our 6-degree-of-freedom rig allows measurement of the baffle reflection characteristics at different incident directions for source and microphone.

Sound measurement setup

The Bell Labs Box is a small $2.19 \times 3.03 \times 2.42$ meter enclosure (a volume of 16.058 meters^3). It comprises a set of configurable panels, a sound source, and a microphone. We constructed it using standard residential housing techniques (see Figure 5).

Sound measurements proceed in two phases. First, in an anechoic chamber, we measure the responses of the sound source, microphone, and surface panels independently. Second, we configure the panels into a room and make measurements of the sound propagation from the source to the microphone at different locations within the room that we later compare to simulation results. (See the next section for more details.)

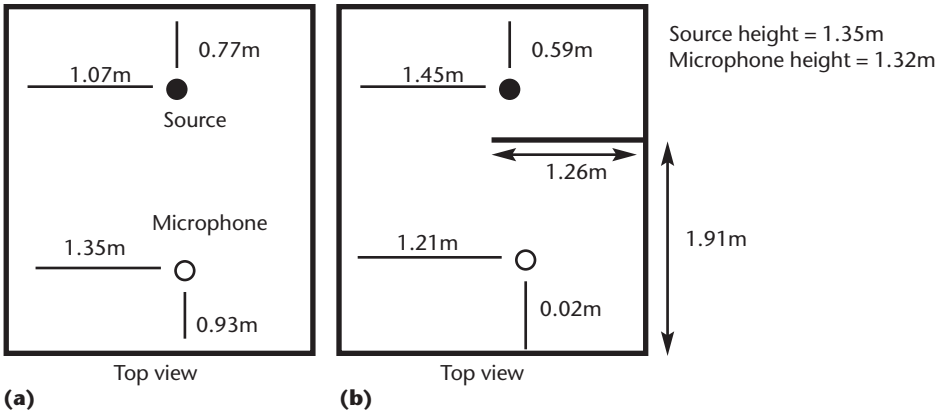
Our sound source is a Brüel & Kjaer artificial mouth type 4227 speaker (http://www.bkhome.com; see Figure 6). During the first phase, we measure its directional responses by placing it on a rotator in the Bell Labs anechoic chamber (http://www.bell-labs.com/org/1133/Research/Acoustics/AnechoicChamber.html) collecting responses at every 5 degrees in the azimuthal plane. (This source has a revolution symmetry by design.)

Our listening device is a Brüel & Kjaer 4134 1/2-inch microphone, calibrated for free-field recording. Based on the manufacturer's specifications, we assume its frequency response is flat, below 10 KHz. We also assume the device is perfectly omnidirectional, below 10 KHz. This defines the upper bound on the frequency for which we can compare measurement and simulations. For directional listening devices, our system would measure their angle-dependent responses using a setup similar to the one we used for sound sources.

We acquire measurements for the sound source and panel responses with a Siglab measurement system (http://www.spectraldynamics.com) using repeated chirp stimuli. We use a 51,200-Hz sampling rate for all measurements.

We made the Bell Labs Box's surface panels out of 3/4-inch melamine, with the smooth surface facing in, fastened to 3/4-inch fire retardant plywood panels. All seams are lapped and caulked. Before assembly, we measure their reflection responses using a bidirectional measurement rig in the anechoic chamber (see Figure 7). The rig has 6 degrees of freedom (three for both source and microphone). Thus, it allows for collecting bidirectional impulse responses of the scattering creat-

ed by the panel. We can use this setup to measure the acoustic characteristics of any material and derive useful parameters, such as the complex acoustic impedance, which can be considered an intrinsic material property for hard surfaces. Using the acoustic impedance, we derive a complex frequency-dependent specular reflection coefficient using the classic model for plane wave reflection, which only depends on the impedance and incident angle.[5]



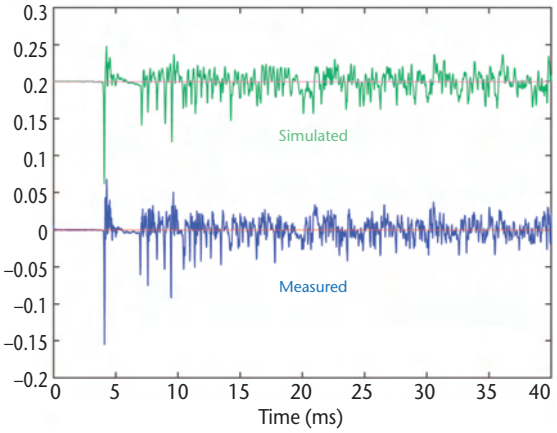
8 **Our two measurement configurations: (a) an empty Bell Labs Box and (b) the Bell Labs Box with a baffle.**

Validation results

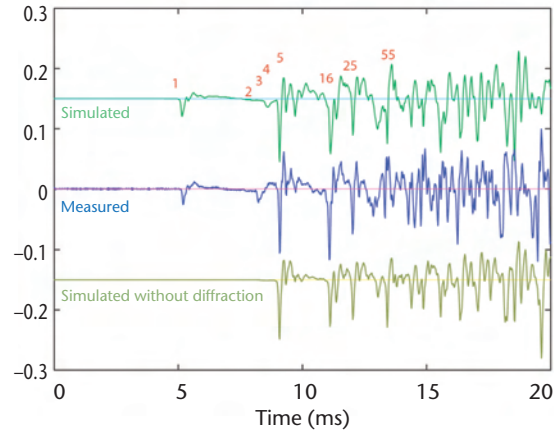
To evaluate our methods, we ran simulations and measurements for two different geometric panel configurations (see Figure 8). The first is a simple box-shaped enclosure comprising six rectangular panels. The second is the same box-shaped enclosure with a rectangular panel spanning from floor to ceiling along one half the box's interior with the speaker and microphone on opposite sides of the panel, as Figure 2 shows. The first configuration is simple, yielding only specular reflections. We use it as a baseline for validation and comparison. The second configuration is a reverberant environment with diffraction, and thus it incorporates propagation paths combining edge diffraction and specular reflection. This is a more difficult case for which the literature hasn't previously provided detailed validation results.

For both configurations, we measured an impulse response using the Brüel & Kjaer artificial mouth and 1/2-inch microphone connected to the audio outputs and inputs of a MOTU 828 multichannel firewire audio interface (http://www.motu.com). We connected the MOTU interface to an off-the-shelf laptop running Windows. The source signal was a repeated chirp stimuli, and the sampling rate was 48 KHz. The output signal used to feed the speaker was also fed back into the interface as a reference. We low-pass filtered the resulting response to get an actual bandwidth of 10 KHz. We simulated all possible propagation paths combining up to the tenth-order specular reflection in the first configuration (empty box), and up to the fourth-order specular reflection and the second-order edge diffraction in the second configuration (with diffracting baffle). These simulation parameters were a reasonable compromise between simulation accuracy and computational expense.

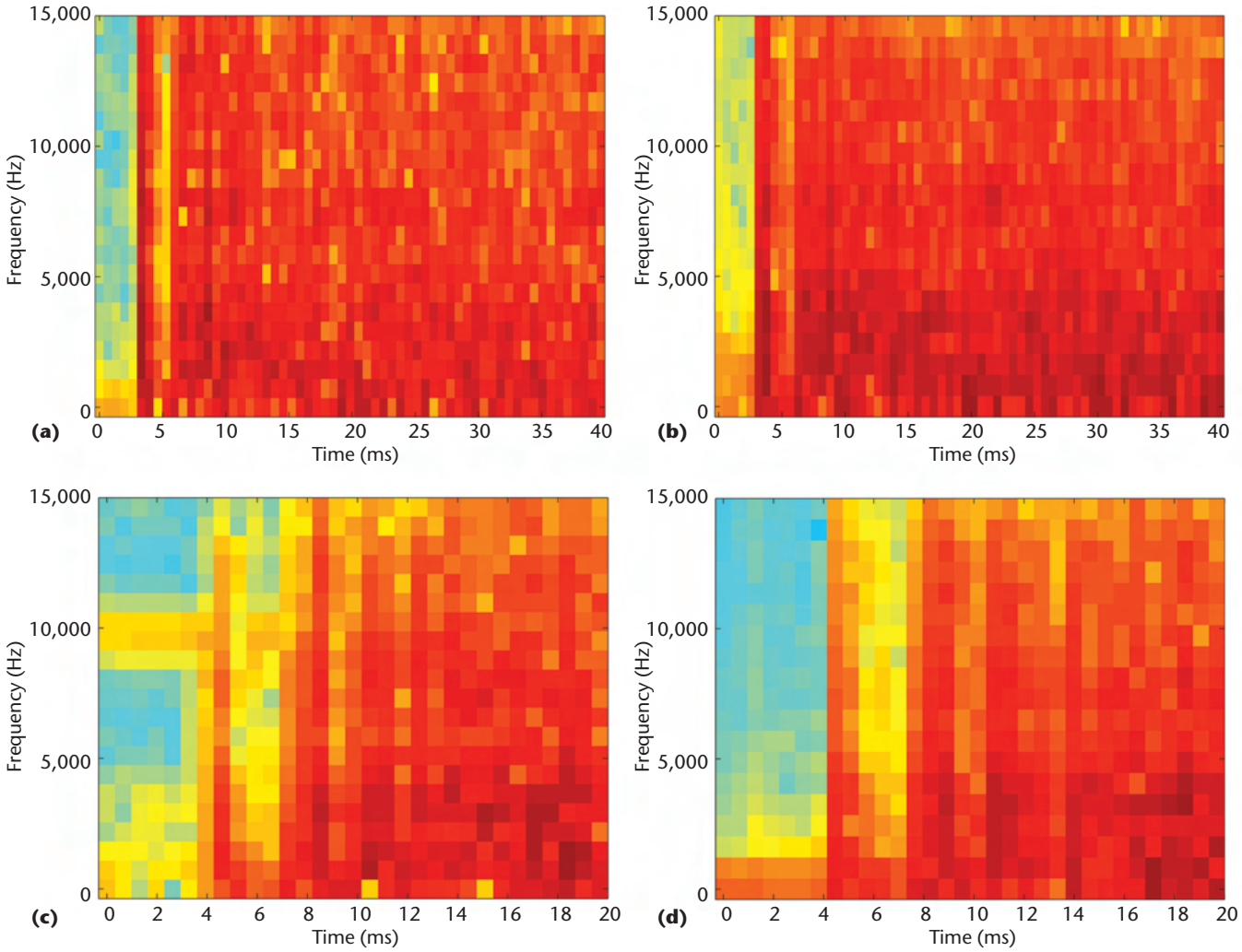
Figures 9 and 10 show the simulated and measured impulse responses for each configuration. Note that fine band simulation with measured source and material characteristics let us compare the waveform of the simulated and measured early responses. As you can see, our simulation can capture the impulse response's temporal structure and can capture the effects of the diffraction by the panel's edge (see the first peak in Figure 10), which is an effect commonly ignored by most acoustic simulation systems. For comparison, we also plotted a simulated impulse response without diffraction effects in Figure 10. Omitting diffraction prevents accurate modeling of the early part of the response but



9 **Comparison of a simulated early impulse response (top) including the first ten orders of specular reflection and a measured response (bottom) in the Bell Labs Box without the baffle. Our simulator computed the contribution of 1,524 propagation paths in 738 seconds.**



10 **Comparison of a simulated early impulse response (top) including the first two orders of diffraction from the edge of the panel and the first four orders of specular reflection and a measured response (middle) in the Bell Labs Box with a baffle. The simulation computed the contribution of 1,358 propagation paths in 631 seconds. The bottom plot shows a simulation including the first eight orders of specular reflection but omitting diffraction (307 paths per 153 seconds).**



11 **Comparison between spectrograms of simulated and measured early responses. The two spectrograms on the top correspond to the (a) measured and (b) simulated impulse responses for the empty box (Figure 9), and those on the bottom correspond to the (c) measured and (d) simulated responses for the box with a baffle (Figure 10).**

has little influence on the later part, which is dominated by specular reflection. A significant amount of energy is absent from the early response when we simulate reflections alone. This can introduce errors when evaluating perceptual criteria based on early to late energy ratios (such as clarity).[14]

Figure 11 shows a comparison between spectrograms for the simulated and measured impulse responses in Figures 9 and 10. From these plots, we can conclude that our simulation performs well for the different frequencies. However, differences appear at low frequencies, where geometrical acoustics provides a poor approximation.

Table 1 shows detailed simulation results for the test configuration with the diffracting baffle. Specifically, the columns list the path ID, length, time delay, and sequence of scattering events for the 60 shortest propagation paths (out of 1,358 simulated). We assigned every face and edge in the 3D model a unique identifier (see Figure 12) to allow detailed analysis of each simulated propagation sequence. For instance, looking in the fourth column of the second row, (s 4) denotes a specular reflection off surface 4 and (d 23) denotes a diffraction off edge 23. The ability to relate the propagation delay to the sequence of events along each path is a powerful tool for analyzing the impulse response because it makes it possible to derive correspondences between features in measured and simulated responses. For instance, we labeled some key features of the simulated response in Figure 10 with the corresponding path identifiers in Table 1. The directly diffracted contribution (the shortest possible path) is clearly identified (#1), followed by contributions of the back and side wall occluded by the panel (#2 and #3). Then, a strong specular reflection out of the other side wall reaches the listener (#5). The strong reflected and diffracted contribution out of the back wall (path #2) is attenuated in our simulation. This might be due to slight inaccuracies in the measured source and microphone positions, which results in interference of this path with the diffracted contribution reflected off the ceiling (path #3). This type of detailed analysis makes it possible to explain discrepancies between our simulation results and our measurements and lets us further improve our simulation models.

Table 1. The 60 shortest propagation sequences (out of 1,358) including four specular reflections and two diffractions off the edge of the panel in the Bell Labs Box with the diffracting baffle.

ID	Path Length (m)	Delay (ms)	Propagation Sequence	ID	Path Length (m)	Delay (ms)	Propagation Sequence
1	1.6758	4.88	(d 23)	31	4.1730	12.15	(s 4) (s 5) (s 6) (d 23)
2	2.7205	7.92	(s 4) (d 23)	32	4.1956	12.22	(d 23) (s 10) (d 23)
3	2.7416	7.98	(d 23) (s 7)	33	4.1956	12.22	(d 23) (s 5) (d 23)
4	3.0022	8.74	(s 5) (d 23)	34	4.1978	12.22	(s 4) (d 23) (s 10)
5	3.0154	8.78	(s 8)	35	4.2772	12.46	(d 23) (s 7) (s 2)
6	3.0603	8.91	(d 23) (s 8)	36	4.3005	12.52	(s 3) (s 1) (d 23)
7	3.1522	9.18	(d 23) (s 9)	37	4.3338	12.62	(s 11) (s 4) (s 6) (d 23)
8	3.1530	9.18	(d 23) (s 10)	38	4.3497	12.67	(s 11) (s 5) (s 4) (d 23)
9	3.3714	9.82	(s 3) (d 23)	39	4.3709	12.73	(d 23) (s 8) (s 2)
10	3.4798	10.13	(s 4) (s 6) (d 23)	40	4.3868	12.77	(s 5) (d 23) (s 8)
11	3.5356	10.30	(d 23) (s 3) (d 23)	41	4.4304	12.90	(d 23) (s 3) (s 1) (d 23)
12	3.5356	10.30	(d 23) (s 8) (d 23)	42	4.4308	12.90	(d 23) (s 10) (s 2)
13	3.5645	10.38	(s 4) (s 5) (d 23)	43	4.4313	12.90	(s 4) (s 3) (s 6) (d 23)
14	3.6860	10.73	(d 23) (s 2)	44	4.4535	12.97	(s 4) (s 5) (s 1) (d 23)
15	3.7042	10.79	(s 5) (s 6) (d 23)	45	4.4766	13.04	(d 23) (s 4) (s 6) (d 23)
16	3.7150	10.82	(s 7) (s 8)	46	4.4795	13.04	(s 5) (d 23) (s 10)
17	3.7515	10.92	(d 23) (s 7) (s 8)	47	4.5513	13.25	(d 23) (s 9) (s 2)
18	3.7515	10.92	(s 11) (s 4) (d 23)	48	4.5804	13.34	(s 4) (d 23) (s 3) (d 23)
19	3.8117	11.10	(s 4) (s 1) (d 23)	49	4.5804	13.34	(s 4) (d 23) (s 8) (d 23)
20	3.8274	11.15	(d 23) (s 7) (s 10)	50	4.5851	13.35	(s 11) (s 4) (s 3) (d 23)
21	3.8637	11.25	(s 4) (s 3) (d 23)	51	4.5872	13.36	(d 23) (s 3) (s 4) (d 23)
22	3.9156	11.40	(d 23) (s 4) (d 23)	52	4.5872	13.36	(d 23) (s 4) (s 3) (d 23)
23	4.0093	11.68	(s 3) (s 6) (d 23)	53	4.6045	13.41	(s 11) (s 4) (s 1) (d 23)
24	4.0176	11.70	(s 5) (s 1) (d 23)	54	4.6433	13.52	(s 4) (d 23) (s 7) (s 8)
25	4.0275	11.73	(s 9) (s 8)	55	4.6964	13.68	(s 4) (s 3) (s 1) (d 23)
26	4.0612	11.83	(d 23) (s 9) (s 8)	56	4.7235	13.75	(d 23) (s 5) (s 6) (d 23)
27	4.1051	11.95	(s 4) (d 23) (s 8)	57	4.7254	13.76	(s 4) (d 23) (s 7) (s 10)
28	4.1315	12.03	(d 23) (s 9) (s 10)	58	4.7308	13.78	(s 4) (d 23) (s 2)
29	4.1483	12.08	(d 23) (s 3) (s 6) (d 23)	59	4.7392	13.80	(d 23) (s 4) (s 1) (d 23)
30	4.1483	12.08	(d 23) (s 8) (s 7) (d 23)	60	4.7561	13.85	(s 3) (d 23) (s 8)

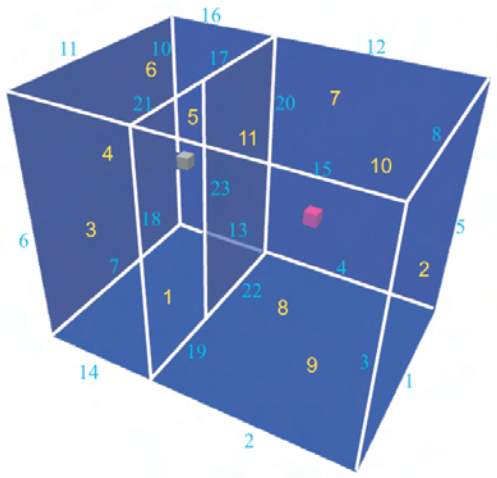
Conclusion and future work

We find that it's possible to achieve remarkably good matches between simulated and measured impulse responses in a carefully controlled experimental environment.

Moreover, because the environment is so simple, we not only can validate the simulations at a gross level (as has been done before) but we can also gain insight into the causes for simulation errors from detailed analysis of the differences between simulated and measured responses. We believe such detailed validation is necessary before we can understand the results of acoustics simulations for more complex environments.

This study is just the beginning of a long path toward understanding the validity of computer-simulated acoustical environments. In the near future, we plan to extend our experiments to more complex environments. Specifically, we intend to consider a series of incremental steps of added complexity (for example, inserting more surfaces, putting boxes in the enclosure, and changing surface materials), measuring and validating the setup after each step. Our general approach is to validate simulations with gradually more complex environments so that we can understand and quantify the limitations of our simulations in detail. Eventually, we hope to validate simulations of concert halls and other more complex real-world environments.

Also, in the near term, we plan to use our experimental setup to evaluate the accuracy of different models for reflection and diffraction in the Bell Labs Box.



12 To track the effect of different propagation sequences on the resulting impulse response, we assign a unique identifier to every edge (cyan) and face (yellow) in our model.

For instance, we could compare the Geometrical Theory of Diffraction[12] with the Biot–Medwin–Tolstoy edge diffraction formulation,[15] which might lead to more accurate simulations at greater computational expense. Or we could evaluate more accurate reflection models, such as Thomasson's exact spherical wave reflection model off planar surfaces.[16] The results of these experiments may lead to an understanding of the trade-offs between computational expense and accuracy of different models.

In the longer term, it will be important to consider psychoacoustical evaluation of acoustic models. We plan to build a system that will use validated acoustic simulations to investigate the psychoacoustic effects of varying acoustic modeling parameters. Our system will let a user interactively change acoustics parameters with real-time auralization and visualization feedback. With such a system, we might be able to address psychoacoustic questions such as

- How many reflections are psychoacoustically important to model?,
- Which surface reflection model provides a psychoacoustically better approximation?, and
- Which among conflicting aural and visual cues are dominant in an interactive virtual environment?

We believe that the answers to such questions are of critical importance to future designers of 3D virtual environments. ■

Acknowledgments

This work was performed while the authors worked at Bell Laboratories. Thomas Funkhouser is partially funded by a National Science Foundation CAREER grant (CCR-0093343) and an Alfred P. Sloan Fellowship. The authors thank the anonymous reviewers for their helpful comments.

References

1. N.I. Durlach and A.S. Mavor, *Virtual Reality Scientific and Technological Challenges*, tech. report, Nat'l Research Council Report, Nat'l Academy Press, 1995.
2. J. Blauert, *Spatial Hearing: The Psychophysics of Human Sound Localization*, MIT Press, Cambridge, Mass., 1997.
3. R.B. Welch, *Perceptual Modification: Adapting to Altered Sensory Environments*, Academic Press, San Diego, 1978.
4. K.R. Boff, L. Kaufman, and J.P. Thomas, eds., *Handbook of Perception and Human Performance*, John Wiley & Sons, New York, 1986.
5. H. Kuttruff, *Room Acoustics*, 3rd ed., Elsevier Applied Science, London, 1991.
6. L.L. Beranek, *Concert and Opera Halls: How They Sound*, American Inst. of Physics, New York,1996
7. T. Whitted, "An Improved Illumination Model for Shaded Display," *C. ACM*, vol. 23, no. 6, June 1980, pp. 343-349.
8. C.M. Goral et al., "Modeling the Interaction of Light Between Diffuse Surfaces," *Computer Graphics* (Proc. Siggraph 84), vol. 18, no. 3, July 1984, pp. 213-222.
9. G.W. Meyer et al., "An Experimental Evaluation of Computer Graphics Imagery," *ACM Trans. Graphics*, vol. 5, no. 1, Jan. 1986, pp. 30-50.
10. T. Funkhouser et al., "A Beam Tracing Approach to Acoustic Modeling for Interactive Virtual Environments," *ACM Computer Graphics, Ann. Conf. Series* (Proc. Siggraph 98), July 1998, pp. 21-32.
11. N. Tsingos et al., "Modeling Acoustics in Virtual Environments Using the Uniform Theory of Diffraction," *ACM Computer Graphics, Ann. Conf. Series* (Proc. Siggraph 01), Aug. 2001, pp. 545-552.
12. J.B. Keller, "Geometrical Theory of Diffraction," *J. Optical Soc. of America*, vol. 52, no. 2, Feb. 1962, pp. 116-130.
13. R.G. Kouyoumjian and P.H. Pathak, "A Uniform Geometrical Theory of Diffraction for an Edge in a Perfectly Conducting Surface," *Proc. IEEE*, vol. 62, Nov. 1974, pp. 1448-1461.
14. L.L. Beranek, *Concert and Opera Halls: How They Sound*, Am. Inst. of Physics, New York, 1996.
15. R. Torres, P. Svensson, and M. Kleiner, "Computation of Edge Diffraction for More Accurate Room Acoustics Auralization," *J. Acoustical Soc. of America*, vol. 109, 2001, pp. 600-610.
16. S.-I. Thomasson, "Reflection of Waves From a Point Source by an Impedance Boundary," *J. Acoustical Soc. of America*, vol. 59, no. 4, Apr. 1976, pp. 780-785.

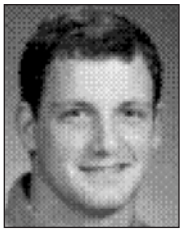
Nicolas Tsingos *holds a tenure research position in the REVES research project at INRIA Sophia Antipolis, France. Previously, he was a member of the technical staff at Bell Laboratories, Lucent Technologies. His research interests are currently in simulating realistic audio for interactive virtual environments, including modeling higher-order phenomena such as sound diffraction. He has a MS and PhD in computer science from the Joseph Fourier University in Grenoble, France. In 2001, he organized and cochaired the Campfire on Acoustic Rendering for Virtual Environments, sponsored by ACM Siggraph and Eurographics.*

Ingrid Carlbom *is the Director of Visual Communications Research in the Multimedia Communications Research Laboratory at Bell Labs. She has a Fil.Kand. from the University of Stockholm, Sweden, a MS in computer science from Cornell University, and a PhD in computer science from Brown University. She has been a director of Siggraph, the chair of the Siggraph Advisory Board, an* IEEE Computer Graphics and Applications *editorial board member, and an* IEEE Transactions on Visualization and Computer Graphics *editorial board member. She is currently a member of the editorial board for* Computers & Graphics *and is coeditor in chief of* Graphical Models. *She has been awarded a Doctor of Philosophy Honoris Causa by Uppsala University,*

Sweden, and is a Distinguished Graduate School Alumna Award by Brown University.

***Gary W. Elko** is president of MH Acoustics in Summit, New Jersey. At Bell Laboratories, he was a supervisor of the acoustics signal-processing group in the Acoustics Research Department. His research interests include eletroacoustic transducer systems, signal processing for spatial audio, and hands-free telecommunication. He has a BSEE from Cornell University and an MS and PhD from the Pennsylvania State University, where he studied acoustics and electrical engineering. He is a past associate editor of the* IEEE Transactions on Acoustics Speech and Signal Processing *and* IEEE Signal Processing Letters *and a fellow of the Acoustical Society of America. Most recently, he was selected as a 2001 IEEE Signal Processing Society distinguished lecturer in the area of communication acoustics signal processing.*

***Robert Kubli** is a partner at MH Acoustics. As a member of Bell Laboratories' Acoustics Research Department, he was responsible for mechanical and electrical design and fabrication of experimental microphone prototypes and test fixtures. He studied mechanical engineering at the Steven Institute of Technology.*



***Thomas Funkhouser** is an assistant professor in the Department of Computer Science at Princeton University. His research interests include interactive visualization of large geometric models, parallel rendering, acoustic modeling, image-based rendering, and 3D mesh analysis. He has a BS in biological sciences from Stanford University, an MS in computer science from the University of California, Los Angeles, and a PhD in computer science from the University of California, Berkeley. He is a member of the IEEE Computer Society.*

Readers may contact Nicolas Tsingos at INRIA Sophia Antipolis, REVES research project, 2004 route des Lucioles, BP 93, F-06902 Sophia Antipolis, France, email Nicolas.Tsingos@sophia.inria.fr.

2002 Editorial Calendar

January/February: Information Visualization

Computer-based information visualization has emerged as a distinct field centered around helping people explore or explain data by designing software that exploits the properties of the human visual system. New methodologies and techniques are critical for helping people keep pace with the torrents of data.

March/April: Image-Based Modeling, Rendering, and Lighting

Despite its recent arrival on the scene, the field of image-based modeling and rendering has already established itself as an important tool for a wide range of computer graphics applications. Image-based techniques use real-world digital photographs to synthesize novel imagery, letting us creatively explore and reinterpret realistic geometry, surface properties, and illumination.

May/June: Graphics in Advanced Computer-Aided Design

The use of computers in the design and manufacturing processes has come a long way from the first CAD systems in the automobile and aerospace industries, with the huge mainframes and enormously expensive displays. Current CAD systems exploit innovative uses of the new technologies that help to move ideas from concept to model to prototype to product.

July/August: Virtual Worlds, Real Sounds

We only need to close our eyes for a moment to experience the amazing variety of information that our ears provide, often more quickly and richly than any other sense. Using real sounds in virtual worlds involves parametric computation; synthesis; and rendering sound for VR, entertainment, and user interfaces.

September/October: Computer Graphics Art History and Archaeology

Archaeologists can use computer graphics techniques to reconstruct and visualize archaeological data of a site that might otherwise be difficult to appreciate, with applications in analysis, teaching, and preservation. Similarly, art historians use computer graphics to analyze, study, and preserve great works of art, which may be too fragile or too valuable to touch or move.

November/December: Tracking

High-resolution tracking of user position and orientation (head, hand, feet, and so on) is increasingly a critical issue for virtual reality, augmented reality, modeling and simulation, and animation. Current tracking hardware is based on a variety of sensors including magnetic, optical, inertial, acoustic, and mechanical (as well as hybrid combinations).

IEEE Computer Graphics AND APPLICATIONS

Eurographics Symposium on Rendering (2007)
Jan Kautz and Sumanta Pattanaik (Editors)

Instant Sound Scattering

N. Tsingos[1], C. Dachsbacher[1], S. Lefebvre[1] and M. Dellepiane[2]

[1]REVES-INRIA, Sophia Antipolis, France
[2]Visual Computing Laboratory, ISTI-CNR, Pisa, Italy

Abstract

Real-time sound rendering engines often render occlusion and early sound reflection effects using geometrical techniques such as ray or beam tracing. They can only achieve interactive rendering for environments of low local complexity resulting in crude effects which can degrade the sense of immersion. However, surface detail or complex dynamic geometry has a strong influence on sound propagation and the resulting auditory perception. This paper focuses on high-quality modeling of first-order sound scattering. Based on a surface-integral formulation and the Kirchhoff approximation, we propose an efficient evaluation of scattering effects, including both diffraction and reflection, that leverages programmable graphics hardware for dense sampling of complex surfaces. We evaluate possible surface simplification techniques and show that combined normal and displacement maps can be successfully used for audio scattering calculations. We present an auralization framework that can render scattering effects interactively thus providing a more compelling experience. We demonstrate that, while only considering first order phenomena, our approach can provide realistic results for a number of practical interactive applications. It can also process highly detailed models containing millions of unorganized triangles in minutes, generating high-quality scattering filters. Resulting simulations compare well with on-site recordings showing that the Kirchhoff approximation can be used for complex scattering problems.

1. Introduction

Proper modeling of sound propagation is very important for virtual acoustics and virtual reality applications. While virtual acoustics has made enormous progress in the recent years, the environments that can be treated interactively remain quite simple. This is mostly due to the the complexity of sound propagation phenomena, in particular the modeling of reflection and diffraction effects. Late reverberation is traditionally rendered by means of statistical techniques or "artificial reverberators" while early scattering is computed using geometrical approaches [FJT02]. Such approaches can model early specular reflection (and possibly edge diffraction) interactively [FCE*98, TFNC01] but are limited to environments of low local complexity. Furthermore, as geometrical acoustics (GA) approaches are valid only for surfaces large compared to the wavelength, it is unclear whether increasing geometrical complexity will provide more accurate results. Hence, surface detail is generally non-existent in GA simulations while it has a strong influence on sound propagation due to the induced scattering. This results in unrealistic effects.

In this paper, we focus on high-quality modeling of first-order sound scattering off complex surfaces using an extended geometrical approach. Our approach handles both diffraction and reflection phenomena in a unified way. Our contributions are:

- The development of an approach to compute the scattering from detailed, dynamic geometry which maps well to programmable graphics hardware (GPUs). This approach is based on a surface integral formulation and uses the Kirchhoff approximation, widely used in acoustics. In particular, we use a *source-view* approach to efficiently find visible surfaces from the source and "mip-mapping" to integrate contributions of all surface samples.

- A level-of-detail approach which reduces the geometry processing for audio rendering while preserving the scattering behaviour of complex surfaces. This is possible thanks to our use of graphics hardware. We believe that this paradigm shift for geometrical acoustics is of prime importance for interactive applications.

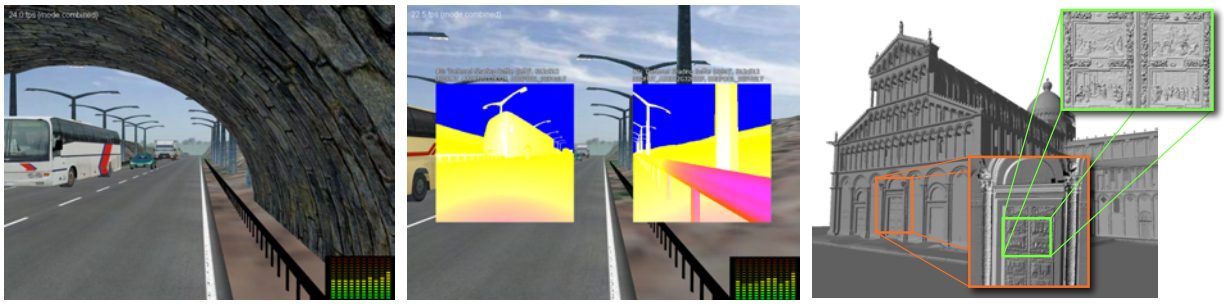


Figure 1: *Left: We present an approach to compute sound scattering effects for interactive applications, including diffraction and direct reflections. Center: Our approach uses the GPU to evaluate a scattering integral on all surfaces visible from a sound source. Right: A 13-million triangle model of the scanned façade of a cathedral and close-ups on surface detail. Our approach can compute perceptually accurate scattering effects over the full audible bandwidth in seconds on such complex, unstructured, models that would be intractable otherwise.*

- An interactive rendering framework for realtime simulation of audio scattering effects which greatly enhance the realism of virtual environments.
- An evaluation of the Kirchhoff approximation for auralization of complex off-line simulations.

While only considering first order phenomena, our approach can provide realistic results for a number of practical interactive applications. It can also handle highly detailed models containing millions of triangles. For such models, we demonstrate that our approach can generate full-bandwidth scattering filters usable for high-quality auralization in minutes. To our knowledge, no other existing approach could be used in practice to process such complex environments.

2. Related Work

A wide variety of methods exist to solve acoustic scattering problems including analytic expansions, approximations and numerical methods [FHLB99]. In the following section we review the techniques most directly related to our work.

Geometrical theories:
Geometrical acoustics (GA) is probably the most widespread approach for interactive acoustic modeling. GA is a high-frequency approximation that models sound propagation along ray-paths. The propagation paths can be constructed using techniques such as ray or beam tracing [LSVA07, FCE*98] using spatial data structures to maximize efficiency. For complex dynamic environments, the cost of updating these structures can become in itself a significant bottleneck. In interactive GA applications, the reflection of sound rays is usually modeled purely specular assuming the size of the surfaces is large compared to the wavelength, which is often not the case in practice. In off-line acoustical simulations, additional lambertian or "glossy" reflections [ISO04,CR05,ZCR06,CDD*06], have been classically used to approximate scattering off complex surfaces, which are then replaced by a flat proxy geometry. However, little work has been devoted to exploring level-of-detail (LOD) approaches in the context of GA [JMT03, WRR04, Sil05], especially how to design general simplification schemes that preserve the correct scattering properties. The framework introduced in this paper brings a possible solution to this issue.

Interactive GA simulations can also be enhanced by introducing diffraction effects from wedges [MPM90] in order to avoid audible discontinuities when the source or a strong specular reflection path becomes occluded from the receiver [TFNC01]. As all GA models, the geometrical theory of diffraction (GTD) assumes edges to be large compared to the wavelength. Increasing geometrical complexity would imply using smaller primitives and eventually would fall outside the validity domain of GA. Experimental studies have shown that low-resolution models might be more appropriate to evaluate acoustical criteria with GA approaches [RSCG99], which is somewhat counter-intuitive. Thus, it is unclear how classical GA+GTD approaches could apply to more realistic scenes, for instance highly detailed models from CAD-CAM or acquired through scanning techniques. Other approaches, such as the Biot-Tolstoy-Medwin model [SFV99,KN00,CS07], follow a Huygens-Fresnel formalism which states that a wavefront can be seen as a superimposition of many elementary "wavelets". The impulse response due to scattering from finite-sized wedges can then be accurately computed by integrating small contributions along the edges. A key aspect of this approach is to directly construct the response in time-domain. However, due to its computational complexity it has found limited use in interactive applications.

Finite and boundary element methods:
Finite element methods (FEM) are numerical solutions to the wave (Helmholtz) equation and associated boundary conditions. They are classically solved in frequency domain by subdividing the environment into small elements (voxels) but alternative time-domain formulations can also be used [SRT94]. Of special interest is the Green surface integral formulation. Using this formulation, the pressure solu-

tion $P(R)$ to the Helmholtz equation can be expressed using an arbitrary surface Σ surrounding the receiver R [FHLB99]:

$$P(R) = P_0(R) - \int_\Sigma (P(U)\nabla G(U,R) - G(U,R)\nabla P(U)) \cdot \mathbf{d}S, \quad (1)$$

where $\mathbf{d}S = \mathbf{n}dS$ ($\mathbf{n}$ unit vector) and $G(U,R) = -e^{ikr}/4\pi r$ is the Green function corresponding to the propagation of a spherical wavefront in free-field (see Figure 2).
$P_0(R)$ is the free-field pressure emitted by the source and the integral term, often called *diffracted field*, is the solution of the homogeneous Helmholtz equation associated with the boundary conditions on the surfaces of the environment. This surface integral is also called the *Helmholtz-Kirchhoff integral theorem*. The Green surface formulation serves as a basis for boundary element methods (BEM) techniques. In this case, Σ corresponds to the surfaces of the environment.

FEM/BEM methods can account for full scattering effects in a unified way and are widely used to compute off-line reference solutions. However, they are not well suited to interactive applications except for very low frequencies since they require a dense subdivision of space or tessellation of surfaces to properly account for interferences. In the case of BEM, edge-length for surface elements must typically be smaller than $1/4th$ of the wavelength. This makes such approaches very time-consuming at high frequencies or for large-scale problems. They also require carefully-designed meshes with uniform elements to limit errors in the solution. They are thus hard to use with most 3D models, particularly those acquired with scanning or designed by CG artists.

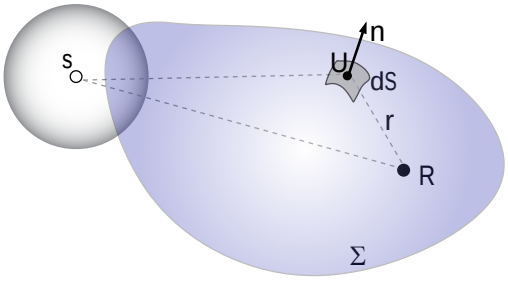


Figure 2: *Notations for the Kirchhoff-Helmholtz integral theorem. S and R denote the source resp. receiver.*

Kirchhoff approximation:
The Kirchhoff approximation (KA) can be seen as a hybrid strategy between GA and wave acoustics [FHLB99]. It is based on Eq. 1 but imposes $P(U) = P_0(U)$ and $\nabla P(U) = \nabla P_0(U)$ on the "illuminated" side of the surfaces (visible to the source) and $P(U) = \nabla P(U) = 0$ on the "shadowed" side. As a result, $P(R)$ can be computed by a direct integration but surfaces facing away from the sound source will not contribute to the solution. However, enforcing the value of both P and ∇P on the same surface is not rigorous for the Helmholtz equation. Hence the obtained result is not a true solution to the wave equation and is generally limited to first-order scattering.

Neglecting the contribution from occluded surfaces and higher-order scattering is the major source of error in the KA. As a result, the approximation will degrade at very low frequencies as the scattered component becomes more important in the regions occluded from the source. The KA will also lead to inaccurate results when second-order occlusions/reflections become prominent. Several studies have shown that it can introduce significant errors in the computed scattering from simple flat or randomly-rough surfaces when compared to reference solutions [JM82, Tho87, NNK93, CL93]. In particular, errors were found to be more important at grazing angles and near-field from the surface. However, some of these studies also used additional far field approximations which might also contribute to the observed errors. As stated in [Tho87, EDS01], further work is still required to evaluate the validity of the KA which is still not well established even today. Despite these limitations, the KA has been widely used to solve off-line scattering/occlusion problems in acoustics [SN81,CI90,CL93,TG98,Emb00,EDS01]. In [SN81] it was shown that the KA can be used to simulate impulse response of first-order reflection/diffraction off rigid panels with good agreement to measured responses. Due to its surface integration and interference treatment, it captures phenomena that cannot generally be modeled using classical GA approaches, providing a continuous first-order sound-field and extended validity range. In this paper, we will also be using the KA and extend the previous work to show that it can be efficiently used to compute very convincing scattering effects off arbitrary complex surfaces.

3. Overview

In Section 4 we show that the Helmholtz-Kirchhoff theorem combined with the Kirchhoff approximation can be used to derive an expression for first-order scattering effects off complex surfaces that compares well with BEM simulations. This formulation is well suited to an implementation using graphics hardware that shares some similarity with the *reflective shadow map* [DS05], introduced to compute interactive first order global illumination effects. Using graphics hardware for audio rendering also opens the opportunity for leveraging classic LOD techniques such as bump or displacement mapping, which we evaluate in Section 5. In Section 6 we present an interactive framework for real-time auralization of complex scattering effects. The throughput of our hardware-accelerated approach makes it well suited for calculating impulse responses of complex architectural environments which can be used for off-line auralization. In this context, Section 7 validates the Kirchhoff approximation against real-life recordings. Finally, in Section 8 we discuss the current limitations of our approach before concluding.

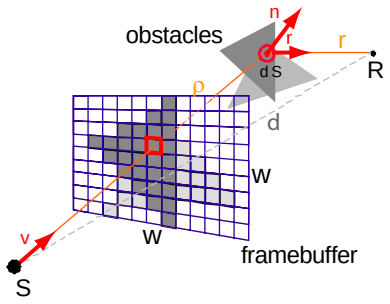


Figure 3: *Surfaces are sampled using hardware rendering from the point of view of the sound source. We evaluate the scattering terms at each pixel before global integration through mip-mapping. In this figure, S and R denote the source resp. receiver.*

4. Scattering from Detailed Geometry

We first derive an expression for first-order scattering off a purely reflecting surface using the Kirchhoff approximation. We assume a steady state harmonic point sound source of frequency f. The free-field pressure created by the source at a listening point R at distance d is given by $P_0(R) = \frac{P_o}{d} e^{ikd}$, where P_o is the pressure amplitude of the source, $k = 2\pi f/c$ and c is the speed of sound (≈ 340 m.s^{-1}). Using the Kirchhoff approximation and following the conventions of Figure 3, the integrand in Eq. 1 becomes:

$$p(R) = -\frac{P_o e^{ik(\rho+r)}}{4\pi\rho r}\left(\left(ik-\frac{1}{\rho}\right)(\mathbf{n}\cdot\mathbf{v})+\left(ik-\frac{1}{r}\right)(\mathbf{n}\cdot\mathbf{r})\right),$$

where $\mathbf{r}$ and $\mathbf{v}$ are unit vectors resp. from the source to the surface-sample and from the surface-sample to the receiver. For additional details, see [Hec98].

Note that this equation does not assume sources or receiver to be in far-field of the scattering surfaces. The diffracted contribution, P_{diff}, received by the listener can thus be interpreted as the difference between the free-field contribution and the portions of the wavefront blocked by the scattering surface Σ:

$$P_{diff}(R) = P_0(R) - \int_\Sigma p(R)dS. \qquad (2)$$

Following the same interpretation, the purely reflected contribution from Σ can be expressed using an integrand $p'(R)$, similar to the one appearing in Eq. 2. In this case, $\mathbf{v}$ is replaced by its mirror-image at the tangent plane. The total scattered pressure, $P(R)$, at the location of the receiver can thus be computed as the sum of the three components, free-field direct, diffracted and reflected as (see Fig. 4 for an illustration):

$$\begin{aligned} P(R) &= P_0(R) + \textstyle\int_\Sigma \left(-p(R)+p'(R)\right)dS, \\ &= P_0(R) + \textstyle\int_\Sigma \hat{p}(R)dS. \end{aligned} \qquad (3)$$

In the more general case of locally-reacting surfaces of finite impedance (i.e., not purely reflecting), the reflection term can be further weighted by the complex-valued reflection coefficient for plane waves [Pie84]. This coefficient will only be valid, however, in far-field from the surface.

Efficient implementation on the GPU

Evaluating Eq. 3 involves two subproblems. First, the integration domain, i.e. all the scattering surfaces visible from the source, must be determined and sampled. In itself, this is a difficult task for complex geometries. Second, the differential contribution of blocked plus reflected wavefronts must be evaluated for all surface samples and summed-up. These two tasks perfectly match the operations implemented by the graphics hardware. To maximize efficiency we implemented these two steps using a "source-view" strategy which provides the most natural mapping to the GPU architecture and results in a very straightforward implementation.

By rendering the scene from the location of the sound source, in a way similar to a *shadow mapping* technique in computer graphics, we can sample the set of directly "illuminated" scattering surfaces (see Figure 3). Computing the source view using perspective projections also provides a form of importance sampling strategy by allocating more fragments to surfaces close to the source.

We can then evaluate the integral in Eq. 3 as a sum over all visible fragments i in this view:

$$\hat{P}(R) \approx \sum_i \hat{p}_i(R)dS_i, \qquad (4)$$

and $dS_i = (w/rez)^2(nearDist/(-\mathbf{u}\cdot\mathbf{t}))^2/(\mathbf{n}\cdot\mathbf{u})$, where $\mathbf{t}$ is the viewing direction, $\mathbf{u}$ the vector from the sample to the viewpoint, $\mathbf{n}$ the normal, *nearDist* is the view plane distance, *rez* the rendering resolution and w the width of the view frustum (assuming aspect ratio is 1). All vectors are unit vectors.

Our GPU implementation renders the geometry to a floating-point offscreen render-target. For each rendered pixel, we evaluate the corresponding value of $\hat{p}_i(R)dS_i$. We store the resulting complex-valued number in two of the four color components of each pixel (see Figure 1 center).

The sum over all visible surfaces can then be efficiently computed using hierarchical integration (i.e., "mip-mapping"), classically performed in $\log(rez)/\log(k)$ render passes, where k is the reduction factor. At each pass a $k \times k$ block of values is summed-up to give a single value which will be recursively integrated in the next pass until the total value of the integral is eventually reached. In our case, we typically used a $4\times$ factor resulting in 5 passes for a 1024×1024 render-target.

To evaluate the integral over multiple frequencies, we use a deferred-shading approach [DS05] and render the necessary geometrical parameters (distances and dot products) only

once. However, Eq. 4 must still be fully re-evaluated. For scenarios where the scattering of multiple dynamic sources must be computed at interactive rates, deferred shading can be leveraged to sample the scattering surfaces from a unique location at the expense of some visibility error. For k sources however, the process requires $k * b$ render passes of the scattering shader (Eq. 4) which currently limits the approach to a small number of sources or frequency components.

Performances

We implemented Eq. 4 both in software using C++ and using programmable graphics hardware with *OpenGL* and *Cg*. For the interactive demos, we used a *Direct3D* implementation. Example shader code can be found at http://www-sop.inria.fr/reves/projects/InstantScattering. We ran performance evaluation tests of our approach on several CPU and GPU configurations using simple models (sphere and plane), where all surfaces are visible from the source. In the software case, sampling the visibility from the source would decrease the performance of the approach for complex geometries. In contrast, using the GPU gives us almost instant access to the visible surfaces. Table 1 summarizes the timings required to compute the full scattering integral *at a single frequency*. Our GPU implementation was found to be 20 to 40 times faster than an equivalent C++ implementation with refresh rates in excess of 700Hz for a fully filled 256×256 render target on our most recent hardware.

Validation against BEM simulations

We first conducted comparisons to BEM simulations using a Fast Multipole Method (FMM) [Dar00] to assess the accuracy and limitations of our approach for a given frequency. We used two simple cases: a spherical occluder made of 8192 triangular elements and a square-shaped plate made of 10200 elements. Figure 4 shows the amplitude of the sound pressure when the receiver rotates around the spherical obstacle. Evaluation was done for 360 positions (every degree). In this case, we obtain very good agreement with the BEM solution although the accuracy of our approach is slightly reduced at low frequencies. Please, refer to the supplemental material at the previously mentionned URL for additional low-frequency comparisons and results for the square-shaped plate.

We also ran comparisons for more complex situations, such as the displacement surface of Figure 5. In this case, second order scattering becomes more prominent and our approach introduces more errors at high frequencies. However, it still provides a reasonable estimation.

5. Simplified Modeling of Surface Detail

Apart from raw computing power, using the GPU also offers the possibility to leverage level-of-detail schemes originally developed for real-time graphics rendering [COM98]. In this

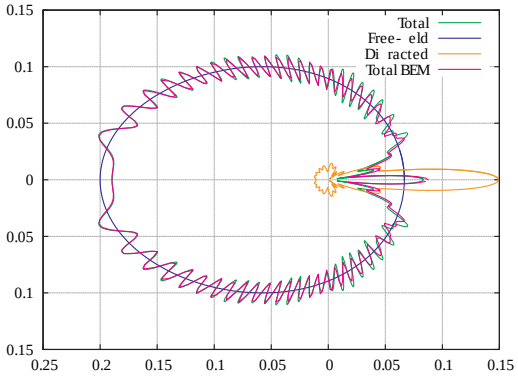


Figure 4: *Polar plot of BEM (magenta) pressure-amplitude solutions compared to our approach (green) as a receiver rotates around a spherical obstacle scattering a 1KHz wave. The radius of the scattering sphere is 1m. Source and receiver are respectively 5 and 10 meters away from the center of the sphere. The plot also shows the unoccluded pressure amplitude (blue) and the amplitude of the scattered component (orange).*

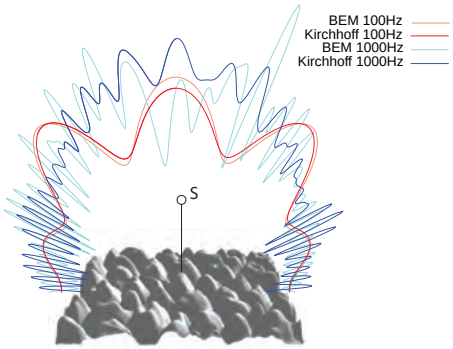


Figure 5: *Scattering patterns for a detailed surface. The figure compares sound pressure levels in a plane medial to the surface obtained by BEM and our approximation. The source is 5m directly above the center of the face and the pressure is plotted at a distance of 10m.*

section, we propose a strategy combining normal mapping with displacement correction to model complex surface detail for acoustic scattering calculations.

Displacement surfaces [CCC87] use textures to encode fine-grain surface detail. They can be rasterized in hardware using a ray-casting approach [HEGD04,BD06], leading to correct visibility handling at the expense of a higher shader-processing cost. To avoid the cost of ray-casting we propose a simpler approach that accounts for the correct propagation delay but omits accurate visibility calculations. To remove the contribution of backfacing fragments, which should not be contributing to the integral, we use an additional weighting term in the integrand of Eq. 4, defined as $-\mathbf{v}.\mathbf{n}$, where $\mathbf{v}$

Hardware	256x256			1024x1024			2048x2048		
	kirch	mipm	total	kirch	mipm	total	kirch	mipm	total
Intel Core2 6600 @2.4GHz	-	-	43	-	-	2.7	-	-	0.7
Intel Xeon @3.2GHz	-	-	35	-	-	2.2	-	-	0.55
QuadroFX Go 1400	380	870	270	37	60	23	n/a	n/a	n/a
GeForce 7950GX2	1120	2120	730	141	170	76	39	26	15

Table 1: *Performance tests on various hardware platforms. The table shows update rates (in Hertz) for the scattering integral computed at single frequency using an increasing number of samples. GPU performance is detailed both for the integrand evaluation shader (kirch) and the summation shader (mipm). Values do not include the cost of sampling the visibility from the source.*

is the direction from the 3D location to the sound source and **n** is the surface normal.

To evaluate our approach, we created several 4x4 meter surface samples from displacement textures. The amplitude of displacement was 0.5 meters. Figure 6 shows example surfaces and scattering impulse responses calculated with true displaced geometry. In this case, source and microphone were directly above the center of the surface respectively 10 and 20 meters away. The impulse responses were computed off-line by evaluating the scattering integral for 8192 linearly spaced frequency components ranging from 0 to 22.05KHz. The result is then obtained by the inverse Fourier transform of the resulting transfer function. A resolution of 1024x1024 was used for the render target. We also created corresponding normal maps from the displaced geometry and performed a scattering calculation using a flat proxy surface with our displacement-corrected normal mapping and standard normal mapping only.

Figure 7 compares the different results. As can be seen, normal maps without displacement correction result in very little difference compared to a flat surface. This demonstrates the importance of the interference phenomena which are paramount in modeling the proper scattering effect. Using our displacement-corrected normal mapping approach results in a much better approximation. We also evaluated our simplification technique at oblique incidence. Please refer to the supplemental materials for an illustration of the results. At grazing incidence, errors have been found to be more significant due to self occlusions but our approximation remains acceptable.

6. Interactive Rendering Framework

Our approach can be used to render interactive scattering effects using a framework similar to the sound occlusion approach of [TG97b], which we extend to include our more physically-grounded scattering calculation.

Our current audio-visual rendering pipeline performs both visual rendering and calculation of the audio scattering coefficients on the GPU. In order to support interactive audio rendering in fully dynamic environments, we compute the scattering integral over a small number of frequencies, typ-

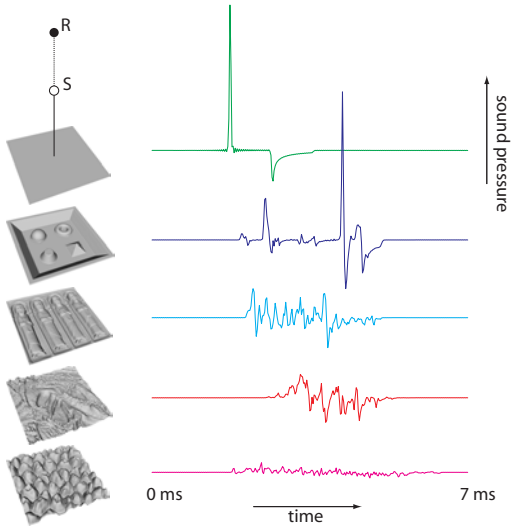


Figure 6: *Responses from different 4×4m surface samples. Each surface is composed of 131072 triangles and generated from displacement maps. Note the secondary scattering component due to the finite extent of the flat surface on the top row (green curve) and the increasingly "diffusing" nature of the surfaces from top to bottom.*

ically 10 to 20, using uniform spacing on a *Bark* scale to provide a better match to auditory perception [Moo97]. This set of frequency subbands b will be selectively modulated by a coefficient defined as:

$$\alpha_b(P) = |P_b(R)|, \tag{5}$$

where $P_b(R)$ is the sum of the direct and scattered components (see Eq. 3) calculated at the center frequency of band b.

All audio processing is performed asynchronously on the CPU. To auralize the effect of scattering we re-equalize the signals based on the coefficients α_b computed on the GPU. We process the input audio signal using short time-frames of 1024 samples with 50% overlap at CD quality. We transform the signal into the Fourier domain and directly multiply each complex coefficient by a scalar factor, which we interpolate from the small set of subband values α_b. This simplified approach does not explicitly model the propagation

delay between the direct and indirect components but will still capture the coloration due to scattering [Hal01]. While appropriate for occlusions, it might result in erroneous rendering of reflections off distant surfaces for which a distinct echo would be perceived. However, psychoacoustical evidence suggests that this approximation will hold for delays up to 50 ms (≈17m difference in path-length) but this threshold generally depends on the level of the reflected sound and the nature of the signal (see [Ber96], Ch. 14). As illustrated in the accompanying video, this approach still results in convincing renderings for a number of practical situations. In particular, we show our technique applied to a prototype driving simulation (see Figure 1) which is a challenging application due to fast moving sources and scatterers of different sizes as well as complex road-side geometry. To demonstrate the flexibility of our approach we compute the scattering of the car's engine from the roadside, tunnels, and other cars, as would be heard through the open windows. The environment in this case comprises 95000 triangles. We used a separate rendering for each side of the car to sample all possible scattering surfaces and provide a stereophonic rendering effect. We also provide examples of interactive auralization of occlusion by a dynamic obstacle and reflections off a complex surface deformed in real-time. The effect of the scattering can be fully appreciated in these examples.

All GPU operations were implemented in *Direct X* and run with frame rates in excess of 100Hz on a *GeForce 8800 GTX*. The GPU could easily handle all the graphics rendering, calculation of the deferred shading buffers and subsequent integration passes for each of the 10 frequencies we used. Note that the graphics rendering also includes per-pixel bump mapping on the road surface and rendering of an equalizer display for illustration purposes. To better balance GPU load between graphics and audio related render passes, we perform calculations for the left and right side of the road every other frame. View-frustum culling was used to optimize rendering but any other acceleration technique can be applied. In our current implementation, 512×512 render targets were used for audio scattering calculations and seemed appropriate to avoid aliasing problems. We believe that the scattered audio component enhances the sense of presence in the environment compared to the direct sound alone since it is tightly coupled to the surrounding geometry. This effect can be appreciated in the accompanying video.

7. Extension to Off-line Simulations

Our approach could also be used to compute high-quality impulse responses suitable for off-line auralization of outdoor acoustic problems. As a first step towards this goal we computed scattering filters for large-scale real-world situations and compared them directly to corresponding recordings in the field. Due to the scale/complexity of the models, we believe that no other approach could be used at this time to obtain such results. Please, refer to the video or the supplemental audio files to evaluate the quality of our simulations.

An interesting example is the Kukulkan temple, a Maya staircase-pyramid located in Chichén Itzá, Mexico (see video and supplemental materials for an illustration). The stairs of this pyramid act as a sound diffraction grating. They reflect a particular chirped echo which has been the object of a number of studies [DDBL04, Bil06]. For additional information, we refer the reader to http://www.ocasa.org/MayanPyramid.htm. We modeled a 856-polygon virtual replica of the pyramid, on which we applied our scattering algorithm. The same model was used for all frequencies and did not require frequency-dependent adaptation of the tessellation.

A full-bandwidth (0-22KHz) transfer function was computed in 2.45 Hz increments (8192 frequencies) in 92 sec. on a *Pentium 4* 3.4GHz workstation using a *Cg/openGL* implementation running on a *GeForce 8800 GTX* GPU. We used a render target of size 1024×1024. In this case, we also included modeling of atmospheric scattering to better account for high-frequency attenuation over long distances. This was simply achieved by weighting the integrand in Eq. 4 by an additional scalar factor computed according to [ISO93]. The obtained filter was then applied to the handclap used in the on-site recording available at the above *URL*. Figure 8 (left) compares the result of our simulation to the recording. Although the recording contains significant environmental noise, it can be seen that our algorithm convincingly captures the chirped echo from the stairs.

Our second, more challenging example models the scattering of the façade of the *Duomo* on the *Piazza dei Miracoli* in Pisa, Italy, also famous for its leaning bell-tower (see Figure 1). We used a detailed model of the cathedral obtained from time-of-flight laser scanning and containing 13 million triangles (a resolution of about 2 cm). Figure 8 (right) compares a simulation with an on-site recording of a handclap. As can be seen, the approach gives satisfying results although some components, probably due to higher-order scattering or reflections from the ground (which was not acquired) are missing. The computing time for the solution was similar to the one of the pyramid since we use a deferred shading approach. Creating the necessary renderings required only 2 additional seconds using the same hardware.

8. Discussion and Limitations

We demonstrated several successful applications of our approach. We now discuss some remaining concerns.

High-frequency and surface aliasing

Our *source-view* approach can be prone to aliasing due to insufficient sampling at high frequencies or to a bad estimation of the dS_i term in Eq. 4. A possible solution to this problem could be to implement the algorithm in surface space by rendering each polygon of the scene in a 2D surface "atlas".

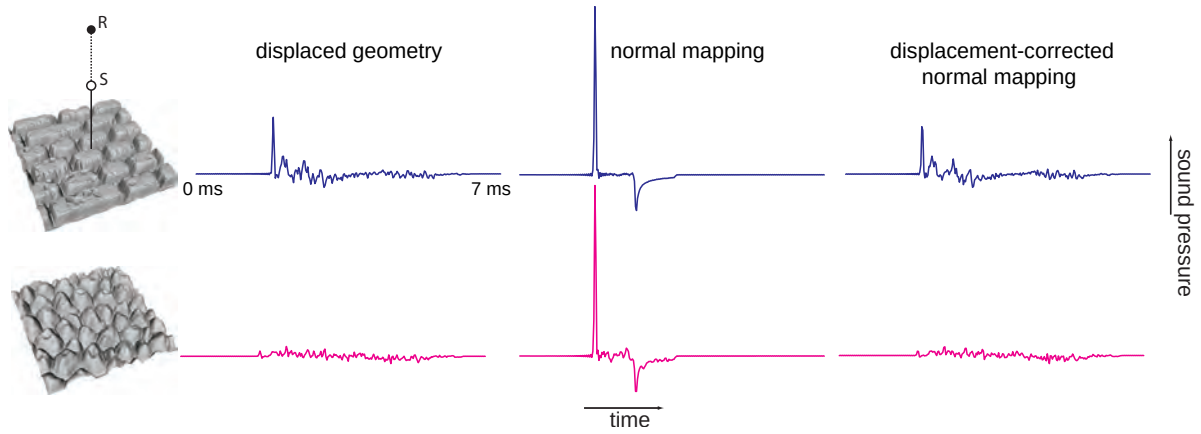


Figure 7: *Comparison of true displaced geometry with a proxy flat quadrilateral enhanced with normal-map only or combined normal/displacement maps. Source and receiver are respectively 10 and 20 m directly above the center of the face. Note how the normal-map alone has little effect on the obtained response.*

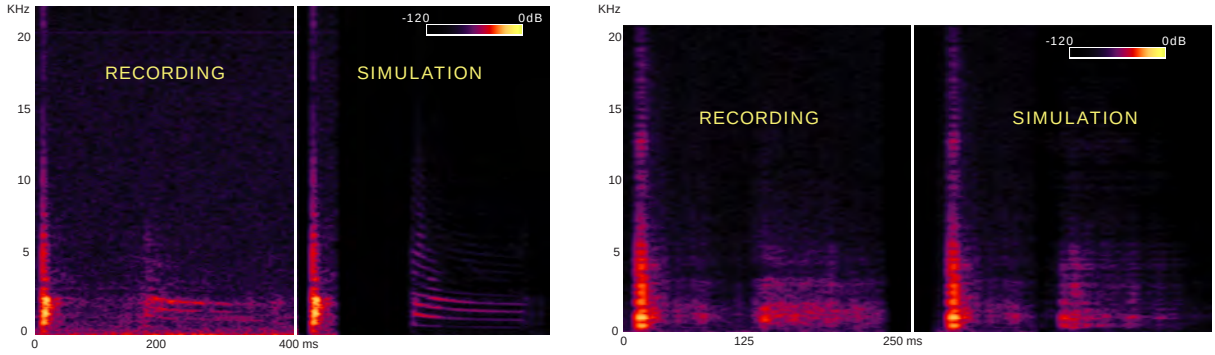


Figure 8: *Comparison between spectrograms of a simulation and an on-site recording for the Kukulkan temple (left) the façade of the Duomo (right). The simulated responses are convolved by the handclap of the original recordings.*

Fragments visible from the source have to be determined through the use of a shadow-map. Such an approach would offer several advantages. First, it would allow for precise control of the surface sampling which could be adapted to each primitive. Second, it could be used to treat omnidirectional sources without requiring six renderings to construct an environment map as it is currently the case with our approach. Although resulting in reduced performance in our preliminary implementation, this alternative approach could be useful for off-line high-quality simulations.

Higher-order scattering

Our approach remains limited to first-order scattering. As a result it cannot be used to compute reverberation. It also does not account for occlusions of first-order reflected waves. This is likely to have an influence at high frequencies. A straightforward solution would be to use an additional shadow map from the receiver to account for self occlusions. However, as can be seen in Figure 5, this would not hold at low frequencies for which it seems better to ignore occlusions. One possible approximation would be to determine a suitable blend between the unoccluded and the occluded solution depending on the frequency. For higher-order propagation between objects, which would be especially relevant for indoor simulations, we believe our approach can be used to enhance GA simulations with realistic precomputed surface scattering functions or filters similar to Figures 5 and 6. This could be achieved in combination with an image-source/beam-tracing technique but would probably better suit a radiosity-like framework to account for non-specular transfer between surfaces. In this context, our approach could also be used to compute the impulse response of the *form-factors* [TG97a].

Simplified frequency-domain processing

Reconstructing an accurate impulse response from sparse data is a very challenging problem which is equally difficult in time or frequency domain. Using a frequency-domain approach is key to our efficient implementation on the GPU since the contribution of all samples can be integrated re-

gardless of their propagation time to the receiver. However, using only a small number of frequency bands introduces an approximation to the true filtering since it makes it difficult to account for phase effects, e.g. due to the propagation delays from the scattering surfaces. For distant surfaces, the delay of the scattered component relative to the direct sound has to be accounted for separately. We believe that an hybrid time-frequency approach could be derived by partitioning the integral into different sub-regions which would contribute separately to the global solution. This could be achieved in world-space by pre-segmenting the scene into different object layers or directly in source-view by stopping our hierarchical integration at a given depth. This latter solution would preserve the efficiency of our current approach. Average or minimum delays could then be computed for each region and used to explicitly account for the propagation time prior to filtering. Spatial audio rendering could also be improved by computing an average direction of incidence to the listener for each region.

9. Conclusion

Our approach casts the problem of sound scattering in a framework based on the GPU for all geometric computations. There are three main advantages to this usage of the GPU. First, the raw power of rasterization, thanks to the massively parallel pipeline which allows the determination of visible surfaces at speeds orders of magnitude faster than previous methods. Any type of primitives that can be rasterized can be used with our approach, for instance direct point-based representations from scanned data. Second, the ability to use a mip-mapping strategy to compute the scattering integral extremely efficiently. Third, the ability to use effective level of detail mechanisms such as displacement mapping to treat extremely complex geometry as textured billboards. We believe that this paradigm shift for geometrical acoustics is of prime importance for interactive applications. Our approach simulates a continuous first-order sound field for complex dynamic environments which would be very difficult, if not impossible, to obtain with concurrent GA techniques. As a result, our approximate interactive rendering allows the addition of convincing and artefact-free sound scattering effects to interactive virtual environments, adding a sense of realism which was previously impossible.

Our GPU-based approach also allows efficient computation of high-quality sound scattering effects at a scale which was previously impossible. Our results show for the first time that the Kirchhoff approximation can be successfully used for off-line auralization in very complex environments. On our simpler validation examples, we could typically compute 32000 integrals during the time required to obtain a BEM solution for the same scene. Hence, improving further on the accuracy of our approach to bring it closer to BEM could be of tremendous interest for off-line acoustics simulations. Finally, with a suitable extension to vibrating surfaces, integration with techniques such as the precomputed acoustic transfer (PAT) [JBP06] would lead to much faster solutions, and would render the PAT approach more attractive.

Acknowledgements

This research was funded by the EU FET Open project IST-014891-2 CROSSMOD (http://www.crossmod.org). C. Dachsbacher received a Marie-Curie Fellowship "ScalableGlobIllum" (MEIF-CT-2006-041306). We thank Autodesk for the donation of *Maya* and G. Sylvand for providing the BEM code used in our comparisons.

References

[BD06] BABOUD L., DÉCORET X.: Rendering geometry with relief textures. In *Graphics Interface '06* (2006).

[Ber96] BERANEK L. L.: *Concert and Opera Halls: How They Sound*. Published for the Acoustical Society of America through the American Institute of Physics, 1996.

[Bil06] BILSEN F. A.: Repetition pitch glide from the step pyramid at Chichen Itza. *J. of the Acoustical Society of America*, 120 (2006), 594.

[CCC87] COOK R. L., CARPENTER L., CATMULL E.: The reyes image rendering architecture. *SIGGRAPH Comput. Graph. 21*, 4 (1987), 95–102.

[CDD*06] COX T., DALENBACK B., D'ANTONIO P., EMBRECHTS J., JEON J., MOMMERTZ E., VÖRLANDER M.: A tutorial on scattering and diffusion coefficients for room acoustic surfaces. *Acta Acustica united with Acustica* (Jul 2006), 1–15.

[CI90] CHEN J., ISHIMARU A.: Numerical simulation of the second-order Kirchhoff approximation from very rough surfaces and a study of backscattering enhancement. *J. Acous. Soc. of America*, 4 (Oct 1990), 1846–1850.

[CL93] COX T., LAM Y.: Evaluations of methods for predicting the scattering from simple rigid panels. *Applied Acoustics 40* (1993), 123–140.

[COM98] COHEN J., OLANO M., MANOCHA D.: Appearance-preserving simplification. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1998), ACM Press, pp. 115–122.

[CR05] CHRISTENSEN C., RINDEL J.: A new scattering method that combines roughness and diffraction effects. *Forum Acousticum, Budapest, Hungary* (2005).

[CS07] CALAMIA P., SVENSSON U.: Fast time-domain edge-diffraction for interactive acoustic simulations. *EURASIP Journal on Applied Signal Processing, special issue on Spatial Sound and Virtual Acoustics* (2007).

[Dar00] DARVE E.: The fast multipole method: numerical implementation. *J. Comp. Physics 160* (2000), 195–240.

[DDBL04] DECLERCQ N. F., DEGRIECK J., BRIERS R., LEROY O.: A theoretical study of special acoustic effects caused by the staircase of the El Castillo pyramid at the maya ruins of Chichen-Itza in Mexico. *J. of the Acoustical Society of America*, 116 (2004), 3328.

[DS05] DACHSBACHER C., STAMMINGER M.: Reflective shadow map. *Proceedings of I3D'05* (2005).

[EDS01] EMBRECHTS J., D.ARCHAMBEAU, STAN G.: Determination of the scattering coefficient of random rough diffusing surfaces for room acoustics applications. *Acta Acustica united with Acustica 87* (June 2001), 482–494.

[Emb00] EMBRECHTS J.: Simulation of first and second-order scattering by rough surfaces with a sound-ray formalism. *J. of Sound and Vibration 229*, 1 (June 2000), 65–87.

[FCE*98] FUNKHOUSER T., CARLBOM I., ELKO G., PINGALI G., SONDHI M., WEST J.: A beam tracing approach to acoustic modeling for interactive virtual environments. *ACM Computer Graphics, SIGGRAPH'98 Proceedings* (July 1998), 21–32.

[FHLB99] FILIPPI P., HABAULT D., LEFEVRE J., BERGASSOLI A.: *Acoustics, basic physics, theory and methods*. Academic Press, 1999.

[FJT02] FUNKHOUSER T., JOT J., TSINGOS N.: Sounds good to me ! Computational sound for Graphics, VR, and Interactive systems. *Siggraph 2002 course #45* (2002).

[Hal01] HALMRAST T.: Sound coloration from (very) early reflections. *ASA, Acoustical Society of America Meeting, Chicago* (June 2001).

[Hec98] HECHT E.: *Optics, Chapter 10, pp. 501-507*. 3rd edition, Addison Wesley, 1998.

[HEGD04] HIRCHE J., EHLERT A., GUTHE S., DOGGETT M.: Hardware accelerated per-pixel displacement mapping. *Proc. of Graphics Interface'04. Canadian Human-Computer Communications Society* (2004), 153–158.

[ISO93] ISO: Acoustics - Attenuation of sound during propagation outdoors - Part 1: Calculation of the absorption of sound by the atmosphere. *International Organization for Standardization, ISO 9613-1* (1993).

[ISO04] ISO: Acoustics - Sound-scattering properties of surfaces - Part 1: Measurement of the random-incidence scattering coefficient in a reverberation room. *International Organization for Standardization, ISO 17497-1* (2004).

[JBP06] JAMES D. L., BARBIČ J., PAI D. K.: Precomputed acoustic transfer: Output-sensitive, accurate sound generation for geometrically complex vibration sources. *ACM Transactions on Graphics (SIGGRAPH 2006) 25*, 3 (Aug. 2006).

[JM82] JEBSEN G., MEDWIN H.: On the failure of the Kirchhoff assumption in backscatter. *J. Acous. Soc. of America*, 5 (Nov 1982), 1607–1611.

[JMT03] JOSLIN C., MAGNENAT-THALMANN N.: Significant fact retreival for real-time 3D sound rendering in complex virtual environments. *Proc. of VRTST 2003* (October 2003).

[KN00] KEIFFER R., NOVARINI J.: A time-domain rough surface scattering model based on wedge diffraction: Application to low-frequency backscattering from two-dimensional sea surfaces. *J. Acous. Soc. of America*, 1 (Jan 2000), 27–39.

[LSVA07] LENTZ T., SCHRÖDER D., VORLÄNDER M., ASSENMACHER I.: Virtual reality system with integrated sound field simulation and reproduction. *EURASIP Journal on Advances in Signal Processing 2007* (2007), Article ID 70540, 19 pages. doi:10.1155/2007/70540.

[Moo97] MOORE B. C.: *An introduction to the psychology of hearing*. Academic Press, 4th edition, 1997.

[MPM90] MCNAMARA D., PISTORIUS C., MALHERBE J.: *Introduction to the Uniform Geometrical Theory of Diffraction*. Artech House, 1990.

[NNK93] NORTON G., NOVARINI J., KEIFFER R.: An evaluation of the Kirchhoff approximation in predicting the axial impulse response of hard and soft disks. *J. Acous. Soc. of America*, 6 (June 1993), 3094–3056.

[Pie84] PIERCE A.: *Acoustics. An introduction to its physical principles and applications*. 3rd edition, pp. 107-111, American Institute of Physics, 1984.

[RSCG99] RINDEL J., SHIOKAWA H., CHRISTENSEN C., GADE A. C.: Comparisons between computer simulations of room acoustical parameters and those measured in concert halls. *Joint meeting of the Acoustical Society of America and the European Acoustics Association, Berlin, 14-19 March* (1999).

[SFV99] SVENSSON U. P., FRED R. I., VANDERKOOY J.: Analytic secondary source model of edge diffraction impulse responses. *J. Acoust. Soc. Am. 106* (1999), 2331–2344.

[Sil05] SILTANEN S.: Geometry reduction in room acoustics modeling. *Master Thesis, Helsinki University Of Technology, Department of Computer Science Telecommunications Software and Multimedia Laboratory* (September 2005).

[SN81] SAKURAI Y., NAGATA K.: Sound reflections of a rigid plane panel and of the "live-end" composed by those panels. *J. Acous. Soc. of Japan*, 1 (Jan 1981), 5–14.

[SRT94] SAVIOJA L., RINNE T., TAKALA T.: Simulation of room acoustics with a 3D finite difference mesh. *Proceedings of Intl. Computer Music Conf. (ICMC94)* (Sept. 1994), 463–466.

[TFNC01] TSINGOS N., FUNKHOUSER T., NGAN A., CARLBOM I.: Modeling acoustics in virtual environments using the uniform theory of diffraction. *ACM Computer Graphics, SIGGRAPH'01 Proceedings* (Aug. 2001), 545–552.

[TG97a] TSINGOS N., GASCUEL J.: A general model for the simulation of room acoustics based on hierarchical radiosity. *Technical sketch, in visual proceedings of SIGGRAPH'97, Los Angeles, USA* (Aug. 1997).

[TG97b] TSINGOS N., GASCUEL J.-D.: Soundtracks for computer animation: sound rendering in dynamic environments with occlusions. *Proceedings of Graphics Interface'97* (May 1997), 9–16.

[TG98] TSINGOS N., GASCUEL J.-D.: Fast rendering of sound occlusion and diffraction effects for virtual acoustic environments. *Proc. 104th Audio Engineering Society Convention, preprint 4699* (May 1998).

[Tho87] THORSOS E.: The validity of the Kirchhoff approximation for rough surface scattering using a gaussian roughness spectrum. *J. Acous. Soc. of America*, 1 (Jan 1987), 78–92.

[WRR04] WANG L., RATHSAM J., RYHERD S.: Interactions of model detail level and scattering coefficients in room acoustic computer simulation. *Intl. Symp. on Room Acoustics, a satelite symposium of ICA, Kyoto, Japan* (2004).

[ZCR06] ZENG X., CHRISTENSEN C., RINDEL J.: Practical methods to define scattering coefficients in a room acoustics computer model. *Applied Acoustics* (2006).

Chapter 5

Perceptual Audio Rendering

Handling 3D audio simulation is a key factor for creating convincing interactive virtual environments. The introduction of auditory cues associated to the different components of a virtual scene together with auditory feedback associated to the user interaction enhances the sense of immersion and presence [HB96, LVK02]. Our spatial auditory perception will be solicited for localizing objects in direction and distance, discriminating between concurrent audio signals and analyzing spatial characteristics of the environment (indoor vs. outdoor contexts, size and materials of the room,...). Typical situations encountered in interactive applications such as video games and simulators require processing of hundreds or thousands of sources, which is several times over the capabilities of common audio dedicated hardware. The main computational bottlenecks are a per sound source cost, which relates to the different effects desired (various filtering processes, Doppler and source directivity simulation, etc.), and the cost of spatialization, which is related to the audio restitution format used (directional filtering, final mix of the different sources, reverberation, etc.). Although a realistic result can be achieved through physical modeling of these steps [Pel01a, LHS01], the processing of complex sound scenes, composed of numerous direct or indirect (reflections) sound sources, can take advantage of perceptually based optimizations in order to reduce both the necessary computer resources and the amount of audio data to be stored and processed. Several auditory perceptual properties may be exploited in order to simplify the rendering pipeline with limited impact on the overall perceived audio quality. The general approach is to structure the sound scene by (1) sorting the relative importance of its components, (2) distributing properly the computer resources on the different signal processing operations and (3) handling the spatial complexity of the scene (Figure 1). These techniques, derived from psycho-acoustics, perceptual audio-coding and auditory scene analysis introduce several concepts similar to those found in computer graphics: selective, progressive and scalable rendering (e.g., visibility/view-frustum culling and geometrical/shading level-of-detail). The following chapter presents an overview of our contributions to such approaches. More details can be found in the related publications:

- Nicolas Tsingos, Emmanuel Gallo and George Drettakis.
  Perceptual Audio Rendering of Complex Virtual Environments.
  Proceedings of ACM SIGGRAPH 2004, August 2004.

- Emmanuel Gallo, Guillaume Lemaitre and Nicolas Tsingos.
  Prioritizing Signals for Selective Real-time Audio Processing.
  Proceedings of ICAD 2005 , Limerick, Ireland, July 2005.

- Nicolas Tsingos.
  Scalable Perceptual Mixing and Filtering of Audio Signals using an Augmented Spectral

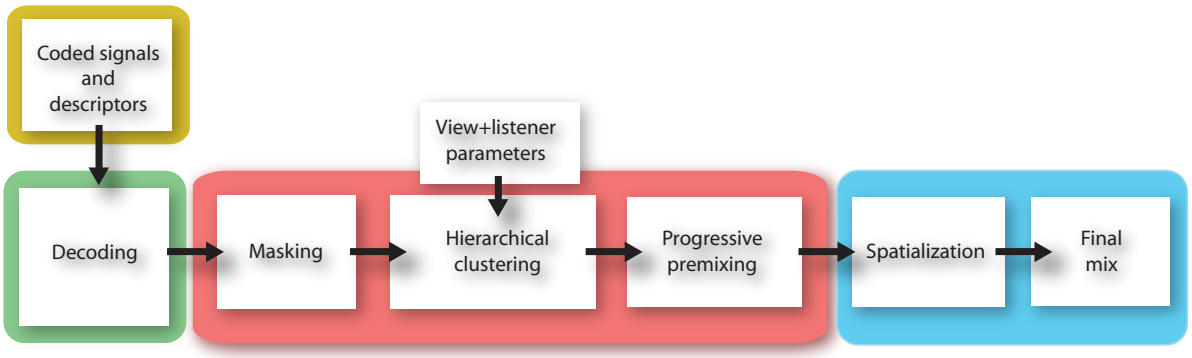


Figure 5.1: Overview of a perceptually-based auralization pipeline for interactive virtual reality applications.

Representation.
Proceedings of DAFX 2005 , Madrid, Spain, September 2005.

- Thomas Moeck, Nicolas Bonneel, Nicolas Tsingos, George Drettakis, Isabelle Viaud-Delmond and David Alloza.
Progressive Perceptual Audio Rendering of Complex Scenes .
Proceedings of the ACM Symposium on Interactive 3D Graphics and Games, 2007.

- Arnault Nagle, Nicolas Tsingos, Guillaume Lemaitre and Aurelien Sollaud.
On-the-fly Auditory Masking for Scalable VOIP Bridges.
Proceedings of the 30th AES Intl. Conf. on Intelligent Audio Environments, 2007.

5.1 Perceptual aspects of spatial audio rendering

5.1.1 Masking and illusory continuity

Selective audio processing approaches build upon prior work from the field of perceptual audio coding that exploits auditory masking. When a large number of sources are present in the environment, it is very unlikely that all will be audible due to masking occurring in the human auditory system [Moo97]. This masking mechanism has been successfully exploited in perceptual audio coding (PAC), such as the well known MPEG I Layer 3 (*mp3*) standard [PS00] and several efficient computational models have been developed in this field. In the context of interactive applications, this approach is thus also linked to the illusion of continuity phenomena [KT02], although current works do not generally include explicit models for this effect. This phenomenon is implicitly used together with masking to discard entire frames of original audio content without perceived artefacts or “holes” in the resulting mixtures.

5.1.2 Importance and saliency of sound sources

Evaluating all possible solutions to the optimization problem required for optimal rendering of a sound scene would be computationally untractable. An alternative is to use greedy approaches which first require estimating the relative importance of each sources in order to get a good starting point. A key aspect is also to be able to dynamically adapt to the content. Several metrics can be used for this purpose such as energy, loudness or the recently introduced saliency. Recent studies have compared some of

these metrics showing that they might achieve different results depending on the nature of the signal (speech, music, ambient sound "textures"). Loudness has been found to be generally leading to better results while energy is a good compromise between complexity and quality.

5.1.3 Limitations of spatial hearing in complex soundscapes

Human spatial hearing limitations, as measured through perceivable distance and angular thresholds [Beg94] can be exploited for faster rendering independently of the subsequent signal processing operations. This is useful for applications where the reproduction format is not set in advance, Recent studies have also shown that our auditory localization is strongly affected in multi-source environments. Localisation performances decrease with increasing number of competing sources [BSK05] showing various effects such as pushing effect (the source localization is repelled from the masker) or pulling effects (the source localization is attracted by the masker) which depend on the time and frequency overlapping between the concurrent sources [BvSJC05]. As a result, spatial simplification can probably be performed even more aggressively as the complexity of the scene, in particular the number of sound sources, grows.

5.1.4 Cross-modal audio-visual interactions

While the primary application of 3D audio rendering techniques is simulation and gaming, no spatial audio rendering work to date evaluates the influence of combined visual and audio restitution on the required quality of the simulation. However, a vast amount of literature in neurosciences suggest that cross-modal effects, such as ventriloquism, might significantly affect 3D audio perception [HWaBS$^+$03, AB04]. This effect tells us that in presence of visual cues, the location of a sound source is perceived shifted toward the visual cue, up to a certain threshold of spatial congruency. Above this threshold, there is a conflict between the perceived sound location and its visual representation and the ventriloquism effect no longer occurs. The spatial window (or angular threshold) of this effect seems to depend on several factors (e.g., temporal synchronicity between the two channels and perceptual unity of the bimodal event) and can vary from a few degrees [LEG01] up to 15° [HWaBS$^+$03]. Research in ventriloquism could imply that we should be more tolerant to localization errors for sound rendering when we have accompanying visuals.

5.2 Algorithms for perceptually-based auralization

5.2.1 Dynamic masking of concurrent sound streams

Interactive applications bring several additional constraints since the masking does not concern a premixed audio stream but has to be evaluated between several concurrent sound signals in course of their processing. Since scenes are generally highly dynamic, masking thresholds have to be continuously predicted and updated according to the instantaneous characteristics of the source signals and their position in space relative to the listener. We introduce dynamic masking of sound sources [TGD04] in order to limit the spatial rendering only to audible sources. The method takes advantage of pre-computed signal characterisation (power spectrum distribution and tonality index [PS00]) associated with each individual audio sample throughout its duration. At runtime, this information is accessed dynamically in order to predict the source's instantaneous loudness [MGB97] anticipating for subsequent frequency-dependent attenuation linked to the source directivity, source-listener distance and possible occlusion or scattering effects. At each frame, sources can thus be sorted according to their loudness contribution at the listener's

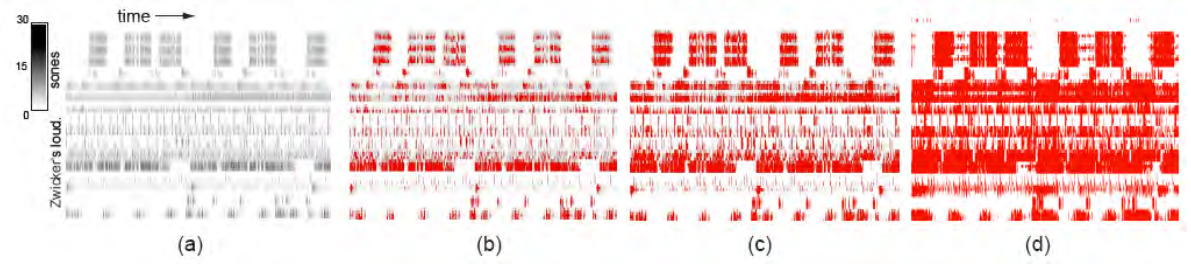


Figure 5.2: Loudness values (using Zwickers loudness model) through time for the 17 tracks of a musical mix. Each track was selectively filtered and processed into 4 frequency sub-bands resulting in 68 signals to prioritize. (b) Priority map showing the first 12.5% most important frames highlighted in red. (c) First 25% most important frames and (d) first 50% most important frames.

ears and summed up until they mask the remaining sources which can then be simply discarded from the rendering pipeline.

5.2.2 Selective and progressive signal processing

When large numbers of sound sources are still present in the scene after the auditory culling (masking) or for systems with limited processing power, per-source processing (e.g., studio-like effects [Zöl02]; Doppler effect; distance, occlusion, reverberation filtering; sub-mixing, etc.) can still represent a strong bottleneck of the audio rendering pipeline. In recent years, several contributions were introduced that aim to bridge the gap between perceptual audio coding and audio processing in order to make audio signal processing pipelines more efficient. Fouad et al. [FHB97] propose a level-of-detail progressive audio rendering approach in the time-domain; by processing every n^{th} sample, artefacts are introduced at low budget levels. Wand and Straßer [MW04] introduce an importance sampling strategy using random selection, but ignore the signal properties, thus potentially limiting the applicability of this method. A family of approaches proposed to directly process perceptually coded audio signals [LS97, TEP04, DDS02] yields faster implementations than a full decode-process recode cycle. In the context of long FIR filtering for reverberation processing, the recent work by Lee et al. [LYLG03] shows that significant improvement can be obtained by estimating whether the result of the convolution is below hearing threshold, hence reducing the processing cost.

Our recent studies have proposed extensions of these approaches by concurrently prioritizing subparts of the original signals to process to guarantee a minimal degradation in the final result [KT02, MBT$^+$07, GLT05, Tsi05b] (see Figure 5.2). Key to such approaches is the choice of a signal representation that allows its progressive encoding and reconstruction. We introduce a progressive signal processing technique where the coefficients of the short-time Fourier transform (STFT) of each signal are pre-computed and stored in decreasing energy order [Tsi05b]. In real-time during the simulation, the algorithm prioritizes the signals and allocates a number of coefficients to process for each source so that a predefined budget of operations is respected. Different perceptual metrics can be used to determine the cut-off point in the list of STFT coefficients, leading to different trade off between computer efficiency and perceptual quality (see Figure 5.3). We complement a loudness-based metrics with a measure of tonality [PS00]. Loud tonal or speech signals, for which the STFT representation is sparser, will require fewer coefficients than a weaker noisier signal for transparent reconstruction. We also compared the merits of this metrics with other models such as the audio saliency map proposed by Kayser et al [KPLL05]. The two metrics lead to very similar results. However, for intermediate budgets and for cases where both

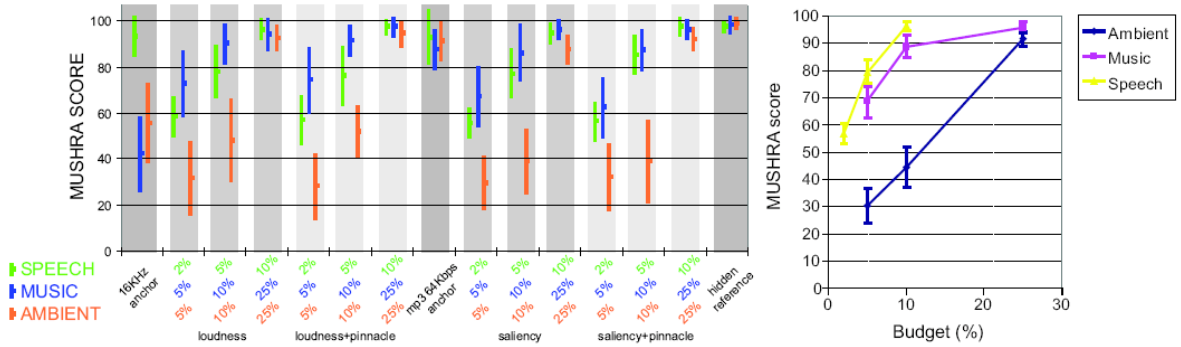


Figure 5.3: Left: Average MUSHRA scores and 95% confidence intervals for our progressive processing tests. Right: Average MUSHRA scores and 95% confidence intervals as a function of budget. Note how perceived quality does not vary linearly with the processing budget and also varies depending on the type (i.e., sparseness) of the sounds.

tonal and noisier signals are used, loudness-based prioritization improves perceived quality relative to the saliency-based alternative (Figure 5.3).

5.2.3 Hierarchical coding of spatial cues in scene-space

One of the primary goals of spatial audio rendering is to reconstruct to the ears of the listeners the perceptual cues which are responsible for localizing sound in direction. The approaches described above can be used to simplify the signal processing operations required by specific 3D audio rendering algorithms. However, it is sometimes desirable to compress the spatial information of the scene in a way independent of the chosen reproduction technique to enable flexible reproduction. In this case, the coding/simplification of the spatial information must be performed in scene-space. This solution which is referred to as scalable spatial audio rendering can be divided into three categories:

- **Fixed basis functions/clustering**: the first set of techniques encodes the spatial cues using a number of fixed basis functions or clusters. For instance, Ambisonics [MM95, DM04] uses a spherical harmonics decomposition of the incoming sound pressure at the listening point. For binaural listening, there are approaches that decompose the Head Related Transfer Functions (HRTFs) onto a basis of eigen-filters corresponding to principal directions [VJGW00, JWL98], and approaches that operate in object space explicitly grouping neighboring sound sources belonging to the same cone of directions [Her99], or using a hierarchical structure [MW04].

- **Dynamic per-object clustering**: The clustering proposed by Sibbald is an object-based method [Sib01]. Sound sources related to an object or an area are grouped according to their distance to the listener. In the near field, secondary sound sources are created and dynamically uncorrelated in order to improve the spatial sensation. In the far field, sources are clustered together, accelerating the spatial rendering. The drawback of the method is that the clustering is evaluated on a per-object basis and does not consider all the elements of the scene.

- **Dynamic global clustering**: We propose a dynamic source clustering method [TGD04] based on both the geometry of the scene and the signals emitted by each source (Figure 5.4). This is especially useful for scenes where sounds are frequently changing in time varying their shape,

energy as well as in location. The algorithm flexibly allocates the required number of clusters; thus clusters are not wasted where they are not needed. The dynamic clustering is derived from the Hochbaum-Shmoys heuristic. The cost-function used for clustering combines instantaneous loudness, distance and angle. An equivalent signal for the cluster is then computed as a mixture of the signals of the clustered sound sources. A representative loudness-weighted centroid is used to spatialize the cluster according to the reproduction setup. This technique has been shown to lead to efficient rendering while maintaining very good rendering quality and minimal impact on localization-task performance, even with a small number of clusters.

5.3 Applications

We know briefly describe how our proposed perceptual rendering techniques can be used in the context of audio rendering for complex auditory scenes and for bandwidth management of audio streamed over networks.

5.3.1 Auralization for interactive virtual environments

Figure 5.1 illustrates the combined use of all previous techniques in the context of a 3D audio rendering engine for complex virtual environments, such as the ones found in video games or simulators. The set of signals for all sound sources are partially decoded so as to retrieve the descriptors and are first tested for masking. Audible sources are clustered and all per-source operations (or premixing) are performed using a progressive approach. Full or scalable decoding of the source signal can be delayed up to this stage, avoiding the cost of streaming/decompressing inaudible signals from the storage media. Finally the obtained signal from each cluster is spatialized using the location of its representative and mixed into the sound output. Note that the framework accommodates both primary sources and secondary sources arising from sound scattering off surfaces.

5.3.2 Concurrent Audio Coding and bandwidth management

The proposed dynamic masking strategy can also be used for bandwidth management, for instance to optimize voice streaming in voice-over-IP applications. In this case, the masking estimation is integrated into a forwarding bridge. Each participant sends coded frames of audio data together with additional descriptors (energy and tonality in a small set of frequency sub-bands) to the forwarding bridge. A dynamic masking procedure is performed for each participant and only the streams audible to each participant are sent downwards. This allows for optimizing bandwidth while maintaining flexible decoding on the client side. Details can be found in [NTLS07] and demos at the following URL: *http://www-sop.inria.fr/reves/OPERA/*

5.4 Discussion

Although our perceptually-motivated approaches use simple models rather than full binaural hearing models, they were found to work very well in practice, as supported by a number of perceptual studies in [GLT05, MBT+07, TGD04, NTLS07] and are already used in commercial game applications (*Test-Drive Unlimited* and *Alone in the Dark* from EdenGames/ATARI). Their simplicity is key in providing a good trade-off between the time required for decision-making in order to optimize resources vs. the cost of the actual processing. Even with the increased computational power of recent multi-core processors,

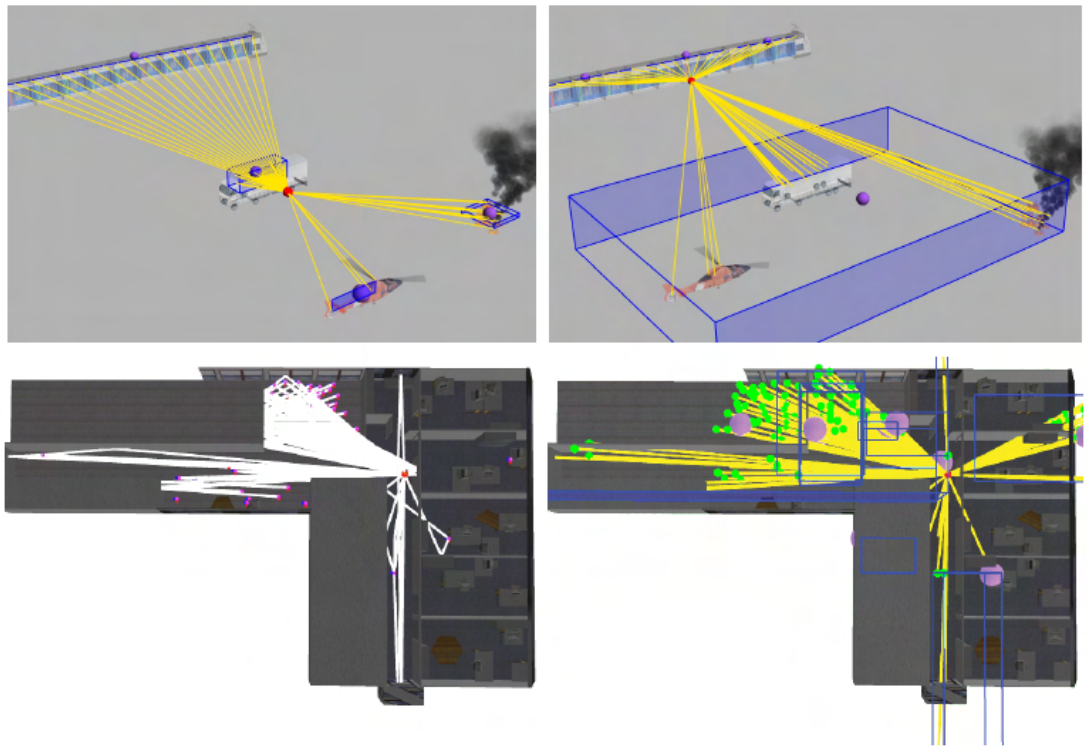


Figure 5.4: Top row: note how the four clusters (in blue) adapt to the listeners location (shown in red). Bottom row: a clustering example with image-sources in a simple building environment (seen in top view). The audible image-sources, shown in green in the right-hand side image, correspond to the set of reflection paths (shown in white) in the left-hand side image.

we believe scalable approaches are still likely to be required in order to tune computing resources or manage bandwidth. They make authoring easier by allowing sound designers to use as many sources as required by the application without thinking about rendering limitations. This is of particular interest for sources linked to physical simulations, such as contact sounds. A strong benefit of these approaches is also to be content-adaptive. An issue is the extra space required for storing additional sound descriptors which might become an issue on systems with limited memory. However, this information can be made quite compact (about 1 KByte per sec. of data) and required storage space should not be a strong limitation for a wide range of platforms. Several extensions could be made to the approaches. First, most approaches for masking or saliency do not account for the 3D location of the sources or use very crude approximations. Spatial unmasking effects inducing variations of masking thresholds due to the relative location of masker and maskee are likely to play a major role in the context of 3D audio perception. Extending current masking approaches to account for this phenomena would be of major interest. Extending the clustering techniques to account for the effect of reverberation would also be of primary interest. Finally, while the main target applications of 3D audio rendering are simulation and gaming, no spatial audio rendering work to date evaluates the influence of combined visual and audio restitution on the required quality of the simulation. However, a vast amount of literature in neurosciences suggests that cross-modal effects, such as ventriloquism, might significantly affect 3D audio perception [HWaBS$^+$03]. We conducted a preliminary study for an extension of our clustering approach to cross-modal phenomena [MBT$^+$07]. However, additional work is still required in this domain.

Perceptual Audio Rendering of Complex Virtual Environments

Nicolas Tsingos, Emmanuel Gallo and George Drettakis
REVES/INRIA Sophia-Antipolis*

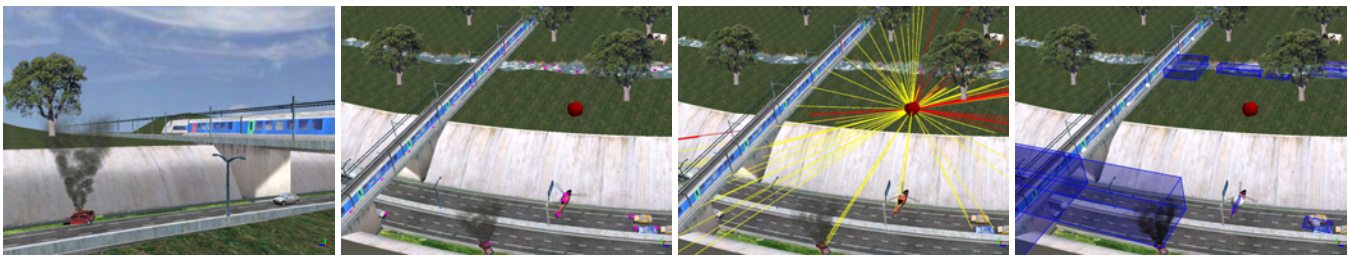


Figure 1: Left, an overview of a test virtual environment, containing 174 sound sources. All vehicles are moving. Mid-left, the magenta dots indicate the locations of the sound sources while the red sphere represents the listener. Notice that the train and the river are extended sources modeled by collections of point sources. Mid-right, ray-paths from the sources to the listener. Paths in red correspond to the perceptually masked sound sources. Right, the blue boxes are clusters of sound sources with the representatives of each cluster in grey. Combination of auditory culling and spatial clustering allows us to render such complex audio-visual scenes in real-time.

Abstract

We propose a real-time 3D audio rendering pipeline for complex virtual scenes containing hundreds of moving sound sources. The approach, based on auditory culling and spatial level-of-detail, can handle more than ten times the number of sources commonly available on consumer 3D audio hardware, with minimal decrease in audio quality. The method performs well for both indoor and outdoor environments. It leverages the limited capabilities of audio hardware for many applications, including interactive architectural acoustics simulations and automatic 3D voice management for video games.

Our approach dynamically eliminates inaudible sources and groups the remaining audible sources into a budget number of clusters. Each cluster is represented by one impostor sound source, positioned using perceptual criteria. Spatial audio processing is then performed only on the impostor sound sources rather than on every original source thus greatly reducing the computational cost.

A pilot validation study shows that degradation in audio quality, as well as localization impairment, are limited and do not seem to vary significantly with the cluster budget. We conclude that our real-time perceptual audio rendering pipeline can generate spatialized audio for complex auditory environments without introducing disturbing changes in the resulting perceived soundfield.

Keywords: Virtual Environments, Spatialized Sound, Spatial Hearing Models, Perceptual Rendering, Audio Hardware.

*contact Nicolas.Tsingos@sophia.inria.fr or visit http://www-sop.inria.fr/reves/

1 Introduction

Including spatialized audio is a key aspect in producing realistic virtual environments. Recent studies have shown that the combination of auditory and visual cues enhances the sense of immersion (e.g., [Larsson et al. 2002]). Unfortunately, high-quality spatialized audio rendering based on pre-recorded audio samples requires heavy signal processing, even for a small number of sound sources. Such processing typically includes rendering of source directivity patterns [Savioja et al. 1999], 3D positional audio [Begault 1994] and artificial reverberation [Gardner 1997; Savioja et al. 1999].

Despite advances in commodity audio hardware (e.g., [SoundBlaster 2004]), only a small number of processing channels (16 to 64) are usually available, corresponding to the number of sources that can be simultaneously rendered.

Although point-sources can be used to simulate direct and low-order indirect contributions interactively using geometric techniques [Funkhouser et al. 1999], a large number of secondary "images-sources" are required if further indirect contributions are to be added [Borish 1984]. In addition, many real-world sources such as a train (see Figure 1) are extended sound sources; one solution allowing their improved, if not correct, representation is to simulate them with a collection of point sources, as proposed in [Sensaura 2001]. This further increases the number of sources to render. This also applies to more specific effects, such as rendering of aerodynamic sounds [Dobashi et al. 2003], that also require processing collections of point sources.

For all the reasons presented above, current state-of-the-art solutions [Tsingos et al. 2001; Fouad et al. 2000; Wenzel et al. 2000; Savioja et al. 1999], still cannot provide high-quality audio renderings for complex virtual environments which respect the mandatory real-time constraints, since the number of sources required is not supported by hardware, and software processing would be overwhelming.

To address this shortcoming, we propose novel algorithms permitting high-quality spatial audio rendering for complex virtual environments, such as that shown in Figure 1. Our work is based on the observation that audio rendering operations (see Figure 2) are usually performed for every sound source while there is significant psycho-acoustic evidence that this might not be necessary due to limits in our auditory perception and localization accuracy [Moore 1997; Blauert 1983].

Similar to the occlusion culling and level of detail algorithms widely used in computer graphics [Funkhouser and Sequin 1993], we introduce a dynamic sorting and culling algorithm and a spatial clustering technique for 3D sound sources that allows for 1) significantly reducing the number of sources to render, 2) amortizing costly spatial audio processing over groups of sources and 3) leveraging current commodity audio hardware for complex auditory simulations. Contrary to prior work in audio rendering, we exploit *a priori* knowledge of the spectral characteristics of the input sound signals to optimize rendering. From this information, we interactively estimate the perceptual saliency of each sound source present in the environment. This saliency metric drives both our culling and clustering algorithms.

We have implemented a system combining these approaches. The results of our tests show that our solution can render highly dynamic audio-visual virtual environments comprising hundreds of point-sound sources. It adapts well to a variety of applications including simulation of extended sound sources and indoor acoustics simulation using image-sources to model sound reflections.

We also present the results of a pilot user study providing a first validation of our choices. In particular, it shows that our algorithms have little impact on the perceived audio quality and spatial audio localization cues when compared to reference renderings.

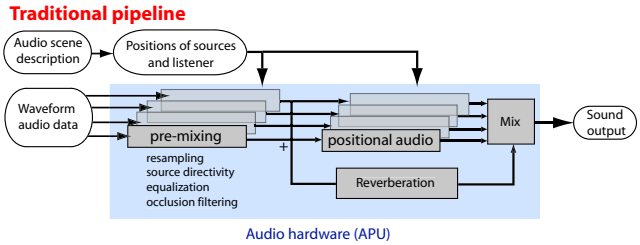


Figure 2: A traditional hardware-accelerated audio rendering pipeline. Pre-mixing can usually be implemented with few operations while positional audio and reverberation rendering require heavier processing.

2 Related Work

Our approach builds upon prior work in the fields of perceptual audio coding and audio rendering. The following sections give a short overview of the background most relevant to our problem.

Perceptual audio coding and sound masking

When a large number of sources are present in the environment, it is very unlikely that all will be audible due to masking occurring in the human auditory system [Moore 1997].

This masking mechanism has been successfully exploited in perceptual audio coding (PAC), such as the well known *MPEG I Layer 3 (mp3)* standard [Painter and Spanias 1997; Brandenburg 1999]. Note that contrary to PAC, our primary goal is to detect masking occurring between several sounds in a dense sound mixture rather than "intra-sound" masking. Since our scenes are highly dynamic, masking thresholds have to be continuously updated. This requires an efficient evaluation of the necessary information.

Interactive masking evaluation has also already been used for efficient modal synthesis [Lagrange and Marchand 2001; van den Doel et al. 2002; van den Doel et al. 2004] but, to our knowledge, no solution to date has been proposed to dynamically evaluate masking for mixtures of general digitally recorded sounds. Such techniques could nevertheless complement our approach for real-time synthesized sounds effects.

In the context of spatialized audio, binaural masking (*i.e.*, taking into account the signals reaching both ears) is of primary importance. Although *mp3* allows for joint-stereo coding, very few PAC approaches aim at encoding spatial audio and include the necessary binaural masking evaluation. This is quite a complex task since binaural masking thresholds are not entirely based on the spatial location of the sources but also depend on the relative phase of the signals at each ear [Moore 1997]. Finally, in the context of room acoustics simulation, several perceptual studies aimed at evaluating masking thresholds of individual reflections were conducted using simple image-sources simulations [Begault et al. 2001]. Unfortunately, no general purpose thresholds were derived from this work.

Spatial audio rendering

Few solutions to date have been proposed which reduce the overall cost of an audio rendering pipeline. Most of them specifically target the filtering operations involved in spatial audio rendering. Martens and Chen et al. [1987; 1995] proposed the use of principal component analysis of Head Related Transfer Functions (HRTFs) to speed up the signal processing operations. One approach, however, optimizes HRTF filtering by avoiding the processing of psycho-acoustically insignificant spectral components of the input signal [Filipanits 1994].

Fouad et al. [1997] propose a level-of-detail rendering approach for spatialized audio where the sound samples are progressively generated based on a perceptual metric in order to respect a budget computing time. When the budget processing time is reached, missing samples are interpolated from the calculated ones. Since full processing still has to be performed on a per source basis, the approach might result in significant degradation for large numbers of sources. Despite these advances, high-quality rendering of complex auditory scenes still requires dedicated multi-processor systems or distributed audio servers [Chen et al. 2002; Fouad et al. 2000].

An alternative to software rendering is to use additional resources such as high-end DSP systems (*Tucker Davis*, *Lake DSP*, etc.) or commodity audio hardware (e.g., *Sound Blaster* [SoundBlaster 2004]). The former are usually high audio fidelity systems but are not widely available and usually support ad-hoc APIs. The latter provide hardware support for game-oriented APIs (e.g., *Direct Sound 3D* [Direct Sound 3D 2004], and its extensions such as *EAX* [EAX 2004]). Contrary to high-end systems, they are widely available, inexpensive and tend to become *de facto* standards. Both classes of systems provide specialized 3D audio processing for a variety of listening setups and additional effects such as reverberation processing. In both cases, however, only a small number of sources (typically 16 to 64) can be rendered using hardware channels. Automatic adaptation to resources is available in *Direct Sound* but is based on distance-culling (far-away sources are simply not rendered) which can lead to discontinuities in the generated audio signal. Moreover, this solution would fail when many sources are close to the listener.

A solution to the problem of rendering many sources using limited software or hardware resources has been presented by Herder [1999a; 1999b] and is based on a clustering strategy. Similar approaches have also been proposed in computer graphics for off-line rendering of scenes with many lights [Paquette et al. 1998]. In Herder [1999a; 1999b], a potentially large number of point-sound sources can be down-sampled to a limited number of representatives which are then used as actual sources for rendering. In theory, such a framework is general and can accommodate primary sound sources and image-sources. Herder's clustering scheme is based on fixed clusters, corresponding to a non-uniform spatial subdivision, which cannot be easily adapted to fit a pre-defined budget. Hence, the algorithm cannot be used as is for resource management purposes. Second, the choice of the cluster representative (the

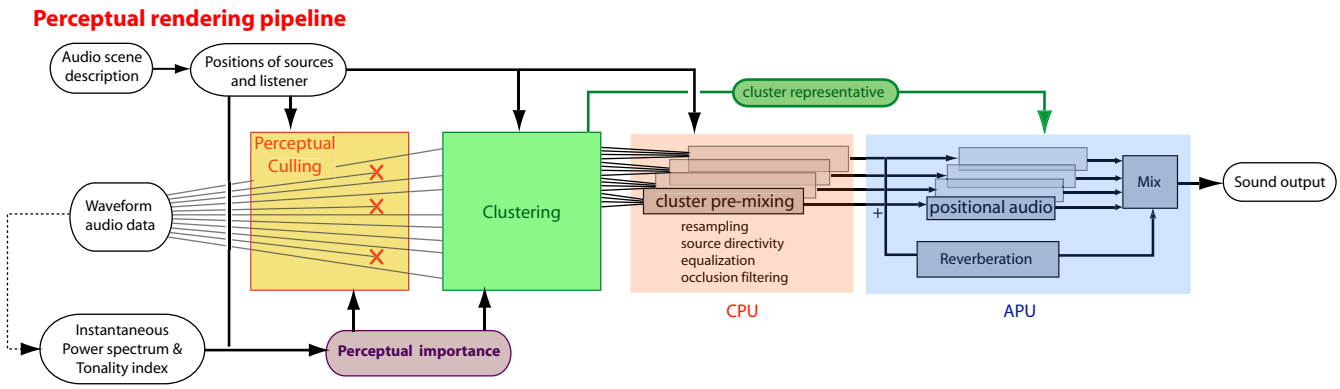


Figure 3: Our novel approach combining a perceptual culling/clustering strategy to reduce the number of sources and amortize costly operations over groups of sound sources.

Cartesian centroid of all sources in the cluster) is not optimal in the psycho-acoustical sense since it does not account for the characteristics of the input audio signals.

3 Overview of our contributions

We propose a novel spatial audio rendering pipeline for sampled sound signals. Our approach can be decomposed into four steps (see Figure 3) repeated for each audio processing frame through time (typically every 20 to 30 milliseconds):

- First, we evaluate the perceptual saliency of all sources in the scene. After sorting all sources based on their binaural *loudness*, we cull perceptually inaudible sources by progressively inserting sources into the mix until their combination masks all remaining ones. This stage requires the pre-computation of some spectral information for each input sound signal.

- We then group the remaining sound sources into a predefined budget number of clusters. We use a dynamic clustering algorithm based on the Hochbaum-Shmoys heuristic [Hochbaum and Schmoys 1985], taking into account the loudness of each source. A representative point source is constructed for each non-empty cluster.

- Then, an equivalent source signal is generated for each cluster in order to feed the available audio hardware channels. This phase involves a number of operations on the original audio data (filtering, re-sampling, mixing, etc.) which are different for each source.

- Finally, the pre-mixed signals for each cluster together with their representative point location can be used to feed audio rendering hardware through standard APIs (e.g., Direct Sound 3D), or can be rendered in software.

Sections 4 to 7 detail each of these steps. We have also conducted a pilot perceptual study with 20 listeners showing that our approach has very limited impact on audio quality and localization abilities. The results of this study are discussed in Section 8.

4 Perceptual saliency of sound sources

The first step of our algorithm aims at evaluating the perceptual saliency of every source. Saliency should reflect the perceptual importance of each source relative to the global soundscape.

Perception of multiple simultaneous sources is a complex problem which is actively studied in the community of auditory scene analysis (ASA) [Bregman 1990; Ellis 1992] where perceptual organization of the auditory world follows the principles of Gestalt psychology. However, computational ASA usually attempts to solve the inverse and more complex problem of segregating a complex sound mixture into discrete, perceptually relevant auditory events. This requires heavy processing in order to segment pitch, timbre and loudness patterns out of the original mixture and remains applicable only to very limited cases.

In our case, we chose the *binaural loudness* as a possible saliency metric. Due to sound masking occurring in our hearing process, some of the sounds in the environment might not be audible. Our saliency estimation accounts for this phenomenon by dynamically evaluating masked sound sources.

4.1 Pre-processing the audio data

In this paper, we focus on applications where the input audio samples are known in advance (*i.e.*, do not come from a real-time input and are not synthesized in real-time). Based on this assumption, we can pre-compute spectral features of our input signals throughout their duration and dynamically access them at runtime.

Specifically, for each input signal, we generate instantaneous short-time *power spectrum distribution (PSD)* and *tonality index* for a number of frequency sub-bands. Such features are widely used in perceptual audio coding [Painter and Spanias 1997].

The PSD measures the energy present in each frequency band, while the tonality index is an indication of the signal noisiness: low indices indicate a noisier component. This index will be used for interactive estimation of masking thresholds.

Our input sound signals were sampled at 44100 Hz. In order to retain efficiency, we use four frequency bands f corresponding to 0-500 Hz, 500-2000 Hz, 2000-8000 Hz and 8000-22050 Hz. Although this is far less than the 25 critical bands used in audio coding, we found it worked well in practice for our application while limiting computational overhead.

We derive our spectral cues from a short time fast Fourier transform (FFT) [Steiglitz 1996]. We used 1024 sample long Hanning-windowed frames with 50% overlap. We store for each band f its instantaneous power spectrum distribution (*i.e.*, the integral of the square of the modulus of the Fourier transform), $\mathbf{PSD}_t(f)$, for each frame t.

From the **PSD**, we estimate a log-scale *spectral flatness measure* of the signal as:

$$\mathbf{SFM}_t(f) = 10\ log10\left(\frac{\mu_g(\mathbf{PSD}_t(f))}{\mu_a(\mathbf{PSD}_t(f))}\right),$$

where μ_g and μ_a are respectively the geometric and arithmetic mean of the PSD over all FFT bins contained in band f. We then estimate the *tonality index*, $\mathbf{T}_t(f)$, as:

$$\mathbf{T}_t(f) = min(\frac{\mathbf{SFM}_t(f)}{-60}, 1).$$

Note that, as a result, $\mathbf{T}_t(f) \in [0,1]$.

This information is quite compact (8 floating-point values per frame, *i.e.*, 1.4 kbyte per second of input audio data at CD quality) and does not result in significant memory overhead.

This pre-processing can be done off-line or when the application is started but can also be performed in real-time for a small number of input signals since our unoptimized implementation runs about six times faster than real-time.

4.2 Binaural loudness estimation

At any given time-frame t of our audio rendering simulation, each source S_k is characterized by : 1) its distance to the listener r, 2) the corresponding propagation delay $\delta = r/c$, where c is the speed of sound, and 3) a frequency-dependent attenuation **A** which consists in a scalar factor for each frequency band. **A** is derived from the octave band attenuation values of the various filters used to alter the source signal, such as occlusion, scattering and directivity filters. For additional information on these filters see [Pierce 1984; ANSI 1978; Tsingos and Gascuel 1997; Savioja et al. 1999]. For instance, in the case of a direct, unoccluded contribution from the source to the receiver, **A** will simply be the attenuation in each frequency band due to atmospheric scattering effects. If the sound is further reflected or occluded, **A** will be obtained as the product of all scalar attenuation factors along the propagation path.

Our saliency estimation first computes the perceptual *loudness* at time t, of each sound source k, using an estimate of the sound pressure level in each frequency band. This estimate pressure level is computed at each ear as:

$$\mathbf{P}_t^k(f) = \mathbf{Spat}(S_k) \times \sqrt{\mathbf{PSD}_{t-\delta}^k(f)} \times \mathbf{A}_t^k(f)/r, \qquad (1)$$

where $\mathbf{Spat}(S_k)$ returns a direction and frequency dependent attenuation due to the spatial rendering (e.g., HRTF processing). In our case, we estimated this function using measurements of the output level of band-passed noise stimuli rendered with *Direct Sound 3D* on our audio board.

As a result, Equation 1 must be evaluated twice since the $\mathbf{Spat}(S_k)$ values will be different for the left and right ear.

The loudness values at both ears $\mathbf{Lleft}_t^k$ and $\mathbf{Lright}_t^k$, are then obtained from the sound pressure levels at each ear using the model of [Moore et al. 1997]. Loudness, expressed in *phons*, is a measure of the *subjective intensity* of a sound referenced to a 1kHz tone[1]. Based on Moore's model, we pre-compute a loudness table for each of our four frequency sub-bands assuming the original signal is a white noise. We use these tables to directly obtain a loudness value per frequency band given the value of $\mathbf{P}_t^k(f)$ at both ears.

Going back to linear scale, a scalar binaural loudness criterion L_t^k is computed as:

$$L_t^k = ||10^{\mathbf{Lleft}_t^k/20}||^2 + ||10^{\mathbf{Lright}_t^k/20}||^2. \qquad (2)$$

Finally, we normalize this estimate and average it over a number of audio frames to obtain smoothly varying values (we typically average over 0.1-0.3 sec. *i.e.*, 4-12 frames).

[1] by definition phons are equal to the sound pressure level, expressed in decibels, of a 1kHz sine wave.

4.3 Binaural masking and perceptual culling

We evaluate masking in a conservative manner by first sorting the sources by decreasing order according to their normalized loudness L_t^k and progressively inserting them into the current mix until they mask the remaining ones.

We start by computing the total power level of our scene

$$\mathbf{P_{TOT}} = \sum_k \mathbf{P}_t^k(f).$$

At each frame, we maintain the sum of the power of all sources to be added to the mix, $\mathbf{P_{toGo}}$, which is initially equal to $\mathbf{P_{TOT}}$.

We then progressively add sources to the mix, maintaining the current tonality $\mathbf{T_{mix}}$, masking threshold $\mathbf{M_{mix}}$, as well as the current power $\mathbf{P_{mix}}$ of the mix. We assume that sound power adds up which is a crude approximation but works reasonably well with real-world signals, which are typically noisy and uncorrelated.

To perform the perceptual culling, we apply the following algorithm, where **ATH** is the absolute threshold of hearing (corresponding to 2 phons) [Moore 1997]:

$\mathbf{M_{mix}} = -200$
$\mathbf{P_{mix}} = 0$
$\mathbf{T} = 0$
$\mathbf{P_{toGo}} = \mathbf{P_{TOT}}$
while $(dB(\mathbf{P_{toGo}}) > dB(\mathbf{P_{mix}}) - \mathbf{M}_{mix})$ **and** $(\mathbf{P_{toGo}} > \mathbf{ATH})$ **do**
  add source S_k to the mix
  $\mathbf{P_{toGo}} -= \mathbf{P}^k$
  $\mathbf{P_{mix}} += \mathbf{P}^k$
  $\mathbf{T} += \mathbf{P}^k * \mathbf{T}^k$
  $\mathbf{T_{mix}} = \mathbf{T}/\mathbf{P_{mix}}$
  $\mathbf{M_{mix}} = (14.5 + \mathbf{Bark}(\mathbf{f_{max}})) * \mathbf{T_{mix}} + 5.5 * (1 - \mathbf{T_{mix}})$
  k++
end

Similar to prior audio coding work [Painter and Spanias 1997], we estimate the masking threshold, $\mathbf{M_{mix}}(f)$ as:

$$\begin{aligned} \mathbf{M_{mix}}(f) &= (14.5 + \mathbf{Bark}(\mathbf{f_{max}})) * \mathbf{T_{mix}}(f) \\ &+ 5.5 * (1 - \mathbf{T_{mix}}(f)) \quad (dB), \end{aligned}$$

where $\mathbf{Bark}(\mathbf{f_{max}})$ is the value of the maximum frequency in each frequency-band f expressed in *Bark* scale. The Bark scale is a mapping of the frequencies in Hertz to Bark numbers, corresponding to the 25 critical bands of hearing [Zwicker and Fastl 1999]. In our case we have for our four bands: $\mathbf{Bark}(500) = 5, \mathbf{Bark}(2000) = 18, \mathbf{Bark}(8000) = 24, \mathbf{Bark}(22050) = 25$.

The masking threshold represents the limit below which a maskee is going to be masked by the considered signal.

To better account for binaural masking phenomena, we evaluate masking for left and right ears and assume the culling process is over when the remaining power at both ears is below the masking threshold of the current mix.

Since we always maintain an overall estimate for the power of the entire scene, our culling algorithm behaves well even in the case of a scene composed of many low-power sources. This is the case for instance with image-sources resulting from sound reflections. A naive algorithm might have culled all sources while their combination is actually audible.

5 Dynamic clustering of sound sources

Sources that have not been culled by the previous stage are then grouped by our dynamic clustering algorithm. Each cluster will act as a new point source representing all the sources it contains (*i.e.*, a point source with a complex impulse response). Our goal is to ensure minimal perceptible error between these auditory impostors and the original auditory scene.

5.1 Building clusters

Sources are grouped based on a distance metric. In our case, we use the sum of two spatial deviation terms from a source S_k to the cluster representative C_n: a distance deviation term and an angular deviation term:

$$d(C_n,S_k) = L_t^k \left(\beta log_{10}(||\mathbf{C_n}||/||\mathbf{S_k}||) + \gamma \frac{1}{2}(1-\mathbf{C_n}.\mathbf{S_k}) \right), \quad (3)$$

where L_t^k is the loudness criterion calculated in the previous section (Eq. 2), $\mathbf{S_k}$ and $\mathbf{C_n}$ are the positions of source S_k and representative C_n expressed in a Cartesian coordinate system relative to the listener's position and orientation.

The weighting term L_t^k ensures that error is minimal for perceptually important sources. In our experiments we used $\beta = 2$ and $\gamma = 1$, to better balance distance and angle errors. Since human listeners perform poorly at estimating distances, our metric is non-uniform in distance space, resulting in bigger clusters for distant sources.

We use a dynamic clustering strategy based on the Hochbaum-Shmoys heuristic [Hochbaum and Schmoys 1985]. In a first pass, this approach selects n potential cluster representatives amongst all k sources by performing a farthest-first traversal of the point set using the metric of Eq. 3. In a second pass, sources are affected to the closest representative, resulting in a disjoint partitioning and clusters are formed. We also experimented with a global $k-means$ approach (e.g., [Likas et al. 2003]), with inferior results in terms of computing time. Both methods, however, gave similar results in terms of overall clustering error (the sum for every source of the distances as defined by Eq. 3).

The representative for the cluster must ensure minimal acoustic distortion when used to spatialize the signal. In particular it must preserve the overall impression of distance and incoming direction on the listener. Thus, a good starting candidate is the centroid of the set of points in (distance, direction) space. Since we are not using a fixed spatial subdivision structure as in [Herder 1999a], the Cartesian centroid would lead to incorrect results for spatially extended clusters. Using the centroid in polar coordinates yields a better representative since it preserves the average distance to the listener.

Moreover, source loudness will affect spatial perception of sound [Moore 1997]. Hence, we use our loudness criterion to shift the location of the representative once the clusters have been determined. The location of the representative is thus defined, in spherical coordinates relative to the listener's location, as:

$$\rho_{C_n} = \frac{\sum_j L_t^j r_j}{\sum_j L_t^j}, \;\; \theta_{C_n} = \theta(\sum_j L_t^j \mathbf{S}_j), \;\; \phi_{C_n} = \phi(\sum_j L_t^j \mathbf{S}_j), \quad (4)$$

where r_j is the distance from source S_j to the listener (S_j's are the sources contained in the cluster).

Figure 4 illustrates the results of our clustering technique in a simple outdoor environment and an indoor environment with sound reflections modeled as image-sources.

5.2 Spatial and temporal coherence

As a result of culling and loudness variations through time, our clustering process might produce different clusters from one frame to another. Since the clusters are mapped one-to-one with audio rendering buffers and the position of a cluster might switch abruptly, audible artefacts might be introduced. To avoid this problem, we perform a consistency check by comparing our current cluster distribution to the one obtained at the previous frame. We shuffle the order of our clusters so that the location of the $i-th$ cluster at frame t is as close as possible to the location of the $i-th$ cluster at frame $t-1$. We sort clusters using the summed loudness of all the sources they contain and perform the test greedily, by comparing distances between all pairs of clusters. Shuffling more perceptually relevant clusters first helps minimize possibly remaining artefacts.

6 Spatial rendering of clusters

The third stage of our pipeline is to compute an aggregate signal (or *pre-mix*) for an entire cluster based on the signals emitted by each individual sound source it contains. This signal will then be spatialized in the final stage of the pipeline.

6.1 Cluster pre-mixing

Computing this pre-mix of all sources involves a number of operations such as filtering (to account for scattering, occlusion, etc.), resampling (to account for the variable propagation delay and Doppler shifting) and $1/r$ distance attenuation of the input signals.

Filtering depends on the source directivity pattern or material properties in case of image-sources. Hence, it should be performed on each source individually. In our case, we use frequency dependent attenuation to account for all filtering effects. We implemented such filtering as a simple "equalization" over our four sub-bands.

For efficiency reasons, we pre-compute four band-passed copies of the original input signals. The filtered signal is then reconstructed as a sum of the band-passed copies weighted by the vector of attenuation values $\mathbf{A}$ (see Section 4.2).

Propagation delay also has to be accounted for on a per source basis. Otherwise, clicking artefacts appear as noticed in [Herder 1999a]. A simple method to account for time-varying non-integer delays is to re-sample the input sound signal. Simple linear interpolation gives good results in practice, especially if the signals are oversampled beforehand [Wenzel et al. 2000].

For maximum efficiency, we implemented these simple operations using SSE (Intel's Streaming SIMD Extensions) optimized assembly code.

Being able to pre-mix each source individually has several advantages. First, we can preserve the delay and attenuation of each source, ensuring a correct distribution of the energy reaching the listener through time. Doing so will preserve most of the spatial cues associated with the perceived size of the cluster and, more importantly, the timbre of the exact mix which would suffer from "comb-filter" effects if a single delay per cluster was used. This is particularly noticeable for reverberations rendered using image-sources. A second advantage of performing pre-mixing prior to audio hardware rendering is that we can provide additional effects currently not (or poorly) supported in existing audio hardware or APIs (e.g., arbitrary directivity patterns, time delays, etc.).

6.2 Spatializing clusters in hardware

The pre-mixed signals for each cluster, along with their representatives, can be used to auralize the audio scene in a standard spatialized audio system.

Each cluster is considered as a point-source located at the position of its representative. Any type of spatial sound reproduction strategy (amplitude panning, binaural processing, etc.) applicable to a point source model can thus be used.

Spatialization can be done in software, limiting the cost of spatial audio processing to the number of clusters. More interestingly, it can be done using standard "game-audio" APIs such as *Direct Sound (DS)*. In this case a hardware 3D audio buffer can be created for each cluster and fed with the pre-mixed signals. The sound buffer is then positioned at the location of the representative

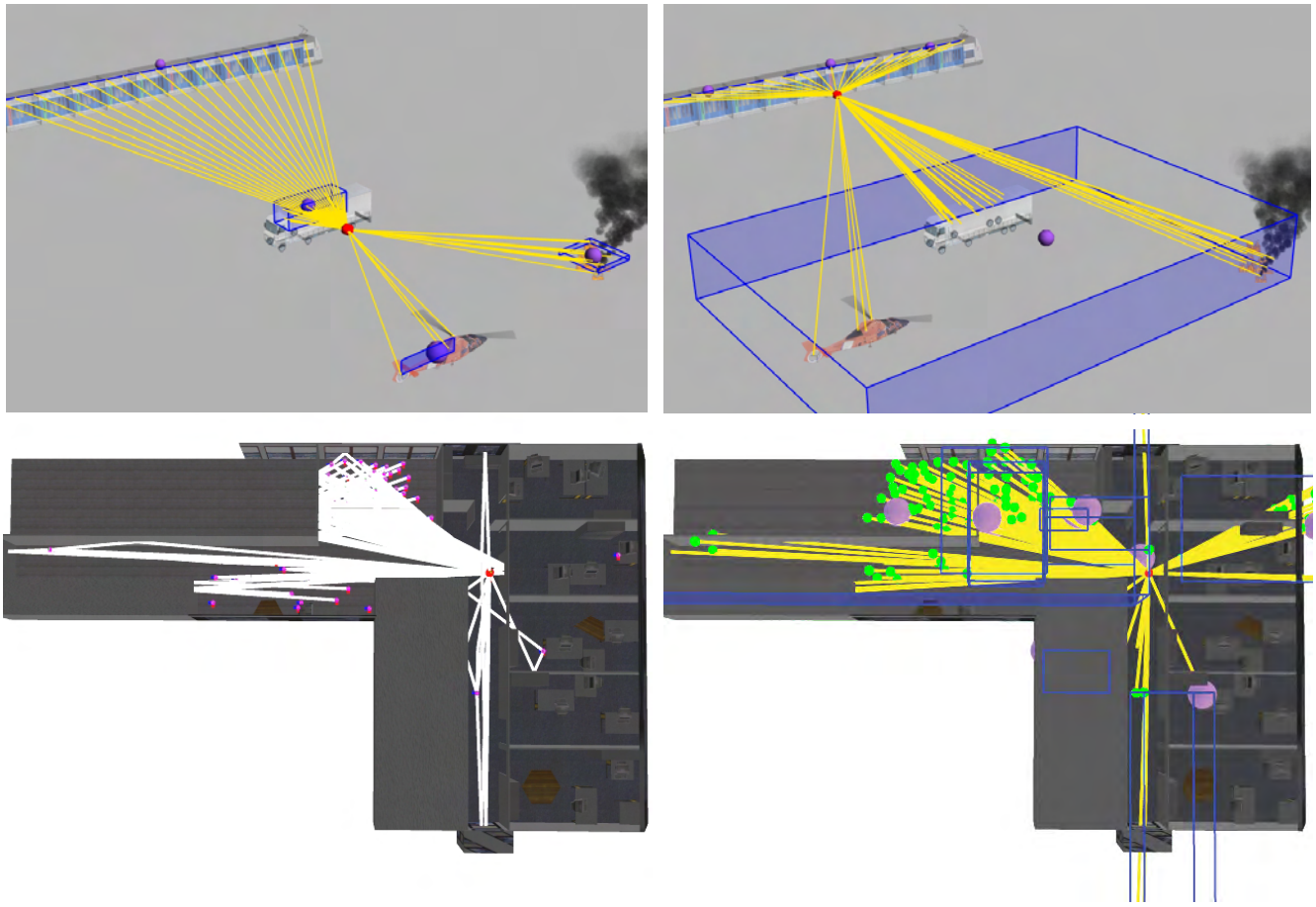


Figure 4: Top row: note how the four clusters (in blue) adapt to the listener's location (shown in red). Bottom row: a clustering example with image-sources in a simple building environment (seen in top view). The audible image-sources, shown in green in the right-hand side image, correspond to the set of reflection paths (shown in white) in the left-hand side image.

(e.g., using DS *SetPosition* command). We synchronize all positional commands at the beginning of each audio frame using DS *notification* mechanism. We also use a simple cross-fading scheme by computing a few extra samples at each frame and blending them with the first samples of the next prior to the spatialization. This eliminates artefacts resulting from sources moving in or out of clusters. In our current implementation, we use a 100-sample overlap at 44.1kHz (*i.e.*, 2ms or about a tenth of our audio frame).

Since audio hardware usually performs sampling-rate conversion, it is also possible to assign a different rate to each cluster depending on its importance. We define the importance of a cluster as the sum of the loudness values of all the sources it contains. We sort the clusters by decreasing importance prior to rendering, and map them to a set of hardware buffers whose sampling rate is decreasing, hence requiring less data to be rendered for an equivalent time-step. This is similar in spirit to the approach of [Fouad et al. 1997] but does not require extra software processing and better integrates with current hardware rendering pipelines.

Finally, we can also benefit from additional processing offered by current consumer audio hardware, such as artificial reverberation processing, as demonstrated in the video (see the trainstation and room acoustics sequences).

7 Applications and performance tests

We evaluated our algorithms on two prototype applications: 1) rendering of one exterior (*Highway*) and one interior (*Trainstation*) scene with numerous point sound sources and 2) rendering of an interior scene (*Busy office*) including modeling of sound reflections.

All tests were conducted on a *Pentium 4 3GHz* PC with a *nVidia GeForce FX5800 ultra* graphics accelerator and a *CreativeLabs Audigy2 platinum Ex* audio board. Audio was rendered using 1200-sample long audio frames at 44.1kHz.

Our first two examples feature *extended* sound sources resulting in many point sources to render. In our case, extended sources are collections of several point sources playing potentially different signals (e.g., the helicopter has 4 sound sources for rotors, jet exhaust and engine, the river is modeled with 40 point sources, etc.). Each sound source can have its own location, orientation and directivity function (e.g., the directivity of the jet exhaust of the helicopter and voice of the pedestrians in the train station are modeled using frequency dependent cosine lobes).

The train station example contains 60 people, with a sound source for their footsteps and one for their voices, two trains, with a source at each wheel, and a number of other sources (pigeons, etc.). A total of 195 sound sources are included. The highway scene contains 100 vehicles and environmental sound effects resulting in 174

environment	#sources	% culled	#clusters	loudness (ms)	culling (ms)	clustering (ms)	pre-mix (ms)	FPS w/o culling (Hz)	FPS w culling (Hz)
Trainstation	195	62	20	1.15	0.42	0.61	2.7	19	27
Highway	174	45	20	1.17	0.42	0.64	2.3	27	33
Busy office	355	71	20	3.8	0.8	1.14	2.5	< 1	22

Table 1: Computing time breakdown for three test environments and corresponding display frame rate (FPS) with and without culling.

Cluster range	All runs							Successful runs		
	localization time (s)			localization error (m)			found (%)	localization time (s)		
	avg.	min.	max.	avg.	min.	max.		avg.	min.	max.
1 to 4	90.40	8.32	408.45	0.49	0.00	4.12	85.2	66.08	8.32	272.86
5 to 7	52.46	7.74	151.31	0.52	0.00	3.16	88.5	34.16	7.74	113.08
8 to 10	66.60	9.18	270.64	0.21	0.00	1.00	100.0	51.98	9.18	127.20
11 to 13	52.43	6.76	204.51	0.32	0.00	2.24	82.1	49.55	6.76	204.51
14 to 16	72.11	8.19	320.82	0.27	0.00	1.00	100.0	61.42	8.19	298.76

Table 2: Statistics for localization time and error for all tests and successful tests only.

sources. The models used for visual rendering contain respectively about 70000 and 30000 polygons and no visibility optimization was used for display.

In the second type of applications we evaluated our rendering pipeline in the context of an interactive room acoustics simulation including the modeling of sound reflection of the walls of the environment. We used a path tracing algorithm to build the image-sources corresponding to sound reflections off the walls of a simple model (a small building floor containing 387 polygons). We simulated all direct paths and first-order reflections to the listener for 60 sound sources resulting in up to 360 image-sources to spatialize. Late reverberation was added to the simulation using the audio hardware's artificial reverberation engine. Note in the video how late reverberation varies as it is driven by the direct and first reflections resulting from our geometrical simulation.

Table 1 summarizes performance results for all sub-parts of our pipeline in the three environments. It also shows averaged video frame rate monitored with and without culling. Performing culling prior to clustering and pre-mixing is a key factor for the efficiency of the algorithm, since the number of sources to process is significantly reduced. In the *busy office* environment, rendering cannot run in real-time if culling is not applied.

Loudness, however, still has to be computed on a per-source basis so its calculation cost becomes significant as the number of sources increases. However, most of the calculation lies in the averaging process necessary to obtain smoothly varying values for the power and tonality estimates. It was our decision to leave this parameter as a variable but we believe averaging could be directly included in the pre-processing step of our approach.

8 Pilot validation study

In order to validate our perceptual rendering pipeline, we conducted a series of experiments aimed at evaluating the impact of culling and clustering sound sources on spatial audio rendering, both in terms of perceived audio quality and localization impairment. We also conducted cross-modal rendering tests to evaluate how visuals impact the perception of our spatial audio rendering.

8.1 Experimental conditions

We conducted our tests using non-individualized binaural presentation over headphones in a quiet office room. For spatial audio rendering, we used *Direct Sound 3D* accelerated by a *CreativeLabs Audigy2 platinum Ex* add-in board on a desktop computer. We used *Sennheiser HD600* headphones, calibrated to a reference listening level at eardrum (using a 1kHz sine tone of maximum amplitude).

The age of our 20 test subjects (12 males/8 females) ranged from 13 to 52 (averaging to 31.8). Most of them were computer scientists but very few having any experience in virtual reality or acoustics.

8.2 Clustering and localization impairment

Our first set of experiments aimed at evaluating the impact of clustering on the spatial perception of the virtual soundscape. In particular, we wanted to evaluate whether localization impairment arises from the clustering process in a task-based experiment. During this test, sound source masking was disabled.

Our experiment is similar in spirit to [Lokki et al. 2000] but uses a different experimental procedure. We asked the test subjects to perform a walkthrough in a 3D environment consisting of many small spheres located every meter on a 2D regular grid at listener's height. Sixteen sound sources corresponding to separate tracks (drums, vocals, guitars, etc.) of a short musical segment were randomly placed at some of these locations. The user was asked to locate an additional reference sound source, emitting a white noise signal, among all the spheres by pointing at it with the mouse and pressing the space bar. Only a single try was allowed. Navigation was done with the mouse, holding the buttons to go forward or backwards. The user could also rotate in place using the arrow-keys of the keyboard.

Each subject performed the test five successive times with a variable number of clusters ranging from small to large. The number of clusters was determined randomly for every test. All subjects underwent a short training period with a reference solution (without clustering) prior to performing the test to learn how to navigate and get accustomed to the 3D audio reproduction.

Subjects performed well in the localization task. Over the 100 runs of the experiment, the source was localized exactly 74% of the time and was found 90% of the time within a 1 meter range of its true location. These results are similar to the ones reported in [Lokki et al. 2000]. More than a half of our subjects localized the source with 100% accuracy. Table 2 reports localization time and error (distance of the selected sphere to the actual target sphere) for the five different cluster ranges. As can be seen, the number of clusters did not have a significant impact on localization time or accuracy.

Music				*Voice*				*Trainstation*									
								cluster rng	with graphics			w/o graphics			both		
cluster rng	avg.	min.	max.	cluster rng	avg.	min.	max.		avg.	min.	max.	avg.	min.	max.	avg.	min.	max.
1 to 4	1.625	-2	4	1 to 8	1.01	-1	3	1 to 8	0.37	0	2	0.35	-1	3	0.36	-1	3
5 to 7	0.433	-1	3	9 to 16	0.7	0	3	9 to 16	0.011	-1	1	0.25	0	2	0.189	-1	2
8 to 10	0.35	-1	3	17 to 24	0.875	0	2	17 to 24	0.364	-0.1	2	0.1	-1	1	0.281	-1	2
11 to 13	0.53	-1	3	25 to 32	0.6	-1	3	25 to 32	0.34	-0.1	2	0.5	0	2	0.42	-0.1	2
14 to 16	0.59	-1	3	33 to 40	0.65	0	3	33 to 40	1.1	-1	4	-0.05	-1	2	0.625	-1	4

Table 3: Statistics for *Reference minus Stimulus* marks for the *Music*, *Voice* and *Trainstation* environments (negative values correspond to cases where the hidden reference received a lower mark than the actual test stimulus).

8.3 Transparency of clustering and culling

The second set of experiments aimed at evaluating the transparency of the combined clustering and culling algorithms on the perceived sound quality.

We used the ITU-R[2] recommended *triple stimulus, double blind with hidden reference* technique, previously used for quality assessment of low bit-rate audio codecs [Grewin 1993]. Subjects were presented with three stimuli, R, A and B, corresponding to the reference, the test stimulus and a hidden reference stimulus[3]. The reference solution was a rendering with a single source per cluster and masking disabled.

Subjects could switch between the three stimuli at any time during the test by pressing the corresponding keys on the keyboard. The reproduction level could also be slightly adjusted around the calibrated listening level until the subject felt comfortable. Subjects were asked to rate differences between each test stimuli (A and B) and the reference R from "imperceptible" to "very annoying", using a scale from 5.0 to 1.0 (with one decimal) [ITU-R 1994].

We used two test environments, featuring different stimulus types. In the first environment (*Music*) the stimulus was a 16-track music signal where each track was rendered from a different location. The locations of the sources were randomized across tests. The second environment (*Voice*) featured a single source with a speech stimulus but also included the 39 first specular reflections from the walls of the environment (a simple shoebox-shaped room). The location of the primary sound source was also randomized across tests.

As before, each subject performed each test five successive times with a variable number of clusters ranging from small to large. The number of clusters was determined randomly for every test.

On average, 63% of the signals were masked during the simulation for the *Music* environment and 33% of the sources were masked in the *Voice* case. Table 3 reports detailed *Reference minus Stimulus* marks averaged over five cluster ranges. Our algorithm was rated 4.4 on average over all tests, very close to the reference according to our subjects (a mark of 4.0 corresponded to *"difference is perceptible but not annoying"* on our scale). This result confirms our hypothesis that our approach primarily trades spatial accuracy and number of sources for computing time while maintaining high restitution quality. For very low cluster budgets (typically 1 to 4) however, significant differences were reported, especially in the *Music* experiment. In such cases cluster locations can vary a lot from frame to frame in an attempt to best-fit the instruments with higher loudness values, resulting in an annoying sensation.

The room acoustics application, while being very well suited to our algorithms, is also very challenging since incorrectly culling image-sources might introduce noticeable changes in the level, timbre or perceived duration of the reverberation. Based on the results of our experiments, our algorithms were found to perform well at preserving the spatial and timbral characteristics of the reverberation in the *Voice* experiment. Our other room-acoustic test (*busy office*, shown in the video) confirms that our algorithm can automatically cull inaudible image-sources while largely preserving the auditory qualities of the reverberation.

Influence of visual rendering

We also attempted to evaluate the influence of possible interaction between image and sound rendering on the quality judgment of our perceptual audio rendering.

We repeated the above quality evaluation test using audio only and audio-visual presentation. Half of our subjects performed the test with visuals and half without. The test environment was a more complex train station environment featuring 120 sources (see video) and we had to limit our maximum number of clusters to 40 to maintain a good visual refresh rate.

Interestingly, the train station example received significantly better marks than the two other examples (see Table 3). This is probably due to its auditory complexity since it contains many simultaneous events, making it harder for the user to focus on a particular problem. For this test-case, the number of subjects was not high enough for a statistical validation of our results. Nonetheless, we note that the obtained marks are lower in the case where visuals were added. This is somewhat counter-intuitive since one could expect that the *ventriloquism effect*[4] would have helped the subjects compensate for any spatial audio rendering discrepancy [Vroomen and de Gelder 2004]. Actually, addition of graphics may have made it easier for the subjects to focus on discrepancies between the audio and visual rendering quality. In particular, some of our subjects specifically complained about having trouble associating voices with the pedestrians. We believe that our simple visual representation of the pedestrians (limited number of models, no facial animation) failed in this case to provide the necessary visual cues to achieve proper cross-modal integration of the voice and faces which is a situation we are highly sensitive to.

This indicates that a cross-modal importance metric should probably be used, possibly increasing the importance of visible sources (as suggested by [Fouad et al. 1997]) and that care should be taken in providing a sufficiently high degree of visual fidelity to avoid disturbing effects in cross-modal experiments.

9 Limitations of our approach

Based on this preliminary user study, our approach seems very promising although the tests were conducted using non-individualized binaural rendering. Using the test subjects' own measured HRTFs might reveal significant differences.

[2] International Telecommunication Union

[3] *i.e.*, the subjects did do not know which of A or B was the actual test or the reference.

[4] "presenting synchronous auditory and visual information in slightly separate locations creates the illusion that the location of the sound is shifted in the direction of the visual stimulus" [Vroomen and de Gelder 2004].

Although the method performs very well for a few hundred sound sources, it cannot easily scale to the thousands due to the cost of the clustering and pre-mixing algorithms. Loudness evaluation for every source would also become a significant bottleneck in this case. More efficient alternatives, such as a simple *A-weighting* of the pressure level could be used and should be evaluated. For such cases, synthesis algorithms might also be used to generate an equivalent signal for the cluster without having to pre-mix all the sources.

Our algorithm currently assumes input sound signals to be non-tonal (noise-like) and uncorrelated. However, the results of the preliminary study indicate that it does perform well on a variety of signals (music, voice, natural and mechanical sounds, etc.). Better results might be achieved by computing a finer estimate of the loudness by combining two values computed assuming the signal is closer to a noise or a tone in each frequency band. This would require determining a representative frequency for each frequency band during the pre-computing step (e.g., the spectral centroid) and using an additional pure-tone loudness table. Loudness values obtained under both assumptions could then be combined using the tonality index to yield a better loudness value. A finer estimate of the pressure level of the current mix and global scene would also improve the masking process. However, this is a more difficult problem for which we do not currently have a solution. Although we perform a sort of temporal averaging when estimating our criteria, we do not account for fine-grain temporal masking phenomena. This is a very interesting area for future research.

Our system currently uses pre-recorded input signals so that necessary spectral information can be pre-computed. However, we do not believe this is a strong limitation. Equivalent information could be extracted during the synthesis process if synthesized sounds are used. Our pre-processing step can also be performed interactively for a small number of real-time acquired signals (e.g., voice acquired from a microphone for telecommunication applications).

Finally, accuracy of the culling and clustering process certainly depends on the number of frequency bands used. Further evaluation is required to find an optimal choice of frequency bands.

10 Conclusion and future work

We presented an interactive auditory culling and spatial level-of-detail approach for 3D audio rendering of pre-recorded audio samples. Our approach allows for rendering of complex virtual auditory scenes comprising hundreds of moving sound sources on standard platforms using off-the-shelf tools. It leverages audio hardware capabilities in the context of interactive architectural acoustics/training simulations and can be used as an automatic 3D voice management scheme in video games. Our current pipeline can render more than ten times the number of sources that could be rendered using consumer 3D audio hardware alone. Hence, future audio APIs and hardware could benefit from including such a management scheme at a lower level (e.g., as part of the *DirectSound* drivers) so that it becomes fully transparent to the user.

We believe our techniques could also be used for dynamic coding and transmission of spatial audio content with flexible rendering, similar to [Faller and Baumgarte 2002]. This would be particularly useful for applications such as massively multi-player on-line games that wish to provide a spatialized "chat room" feature to their participants.

A pilot validation study shows that degradation in audio quality as well as localization impairment is very limited and does not seem to significantly vary with the number of used clusters.

We are currently preparing a full-blown follow-up study to provide additional statistical evaluation of the impact of our algorithm. Further experiments also need to be designed in order to evaluate several additional factors such as the pitch, beat or similarity of the signals in the culling and clustering process.

Our results so far suggest that spatial rendering of complex auditory environments can be heavily simplified without noticeable change in the resulting perceived soundfield. This is consistent with the fact that human listeners usually attend to one perceptual stream at a time, which stands out from the background formed by other streams [Moore 1997].

Acknowledgments

This research was supported in part by the *CREATE* 3-year RTD project funded by the 5th Framework Information Society Technologies (IST) Program of the European Union (IST-2001-34231), http://www.cs.ucl.ac.uk/create/. The authors would like to thank Alexandre Olivier for the modeling and texturing work. The train station environment in the video was designed, modeled and animated by Yannick Bachelart, Frank Quercioli, Paul Tumelaire, Florent Sacré and Jean-Yves Regnault. We thank Alias|***wavefront*** for the generous donation of their **Maya** software. Finally, the authors would like to thank Mel Slater for advice regarding the pilot validation study, Agata Opalach for thoroughly proof-reading the paper and the anonymous reviewers for their helpful comments.

References

ANSI. 1978. American national standard method for the calculation of the absorption of sound by the atmosphere. *ANSI S1.26-1978, American Institute of Physics (for Acoustical Society of America), New York*.

BEGAULT, D. R., MCCLAIN, B. U., AND ANDERSON, M. R. 2001. Early reflection thresholds for virtual sound sources. In *Proc. 2001 Int. Workshop on Spatial Media*.

BEGAULT, D. R. 1994. *3D Sound for Virtual Reality and Multimedia*. Academic Press Professional.

BLAUERT, J. 1983. *Spatial Hearing : The Psychophysics of Human Sound Localization.* M.I.T. Press, Cambridge, MA.

BORISH, J. 1984. Extension of the image model to arbitrary polyhedra. *J. of the Acoustical Society of America 75*, 6.

BRANDENBURG, K. 1999. mp3 and AAC explained. *AES 17th International Conference on High-Quality Audio Coding* (Sept.).

BREGMAN, A. 1990. *Auditory Scene Analysis, The perceptual organization of sound*. The MIT Press.

CHEN, J., VEEN, B. V., AND HECOX, K. 1995. A spatial feature extraction and regularization model for the head-related transfer function. *J. of the Acoustical Society of America 97* (Jan.), 439–452.

CHEN, H., WALLACE, G., GUPTA, A., LI, K., FUNKHOUSER, T., AND COOK, P. 2002. Experiences with scalability of display walls. *Proceedings of the Immersive Projection Technology (IPT) Workshop* (Mar.).

DIRECT SOUND 3D, 2004. Direct X homepage, Microsoft©. http://www.microsoft.com/windows/directx/default.asp.

DOBASHI, Y., YAMAMOTO, T., AND NISHITA, T. 2003. Real-time rendering of aerodynamic sound using sound textures based on computational fluid dynamics. *ACM Transactions on Graphics 22*, 3 (Aug.), 732–740. (Proceedings of ACM SIGGRAPH 2003).

EAX, 2004. Environmental audio extensions 4.0, Creative©. http://www.soundblaster.com/eaudio.

ELLIS, D. 1992. A perceptual representation of audio. *Master's thesis, Massachusets Institute of Technology*.

FALLER, C., AND BAUMGARTE, F. 2002. Binaural cue coding applied to audio compression with flexible rendering. In *Proc. 113th AES Convention*.

FILIPANITS, F. 1994. Design and implementation of an auralization system with a spectrum-based temporal processing optimization. *Master thesis, Univ. of Miami*.

FOUAD, H., HAHN, J., AND BALLAS, J. 1997. Perceptually based scheduling algorithms for real-time synthesis of complex sonic environments. *proceedings of the 1997 International Conference on Auditory Display (ICAD'97), Xerox Palo Alto Research Center, Palo Alto, USA*.

FOUAD, H., BALLAS, J., AND BROCK, D. 2000. An extensible toolkit for creating virtual sonic environments. *Proceedings of Intl. Conf. on Auditory Display* (Atlanta, USA, May 2000).

FUNKHOUSER, T., AND SEQUIN, C. 1993. Adaptive display algorithms for interactive frame rates during visualization of complex virtual environments. *Computer Graphics (SIGGRAPH '93 proceedings), Los Angeles, CA* (August), 247–254.

FUNKHOUSER, T., MIN, P., AND CARLBOM, I. 1999. Real-time acoustic modeling for distributed virtual environments. *ACM Computer Graphics, SIGGRAPH'99 Proceedings* (Aug.), 365–374.

GARDNER, W. 1997. Reverberation algorithms. In *Applications of Digital Signal Processing to Audio and Acoustics*, M. Kahrs and K. Brandenburg, Eds. Kluwer Academic Publishers, 85–131.

GREWIN, C. 1993. Methods for quality assessment of low bit-rate audio codecs. *proceedings of the 12th AES conference*, 97–107.

HERDER, J. 1999. Optimization of sound spatialization resource management through clustering. *The Journal of Three Dimensional Images, 3D-Forum Society 13*, 3 (Sept.), 59–65.

HERDER, J. 1999. Visualization of a clustering algorithm of sound sources based on localization errors. *The Journal of Three Dimensional Images, 3D-Forum Society 13*, 3 (Sept.), 66–70.

HOCHBAUM, D. S., AND SCHMOYS, D. B. 1985. A best possible heuristic for the *k*-center problem. *Mathematics of Operations Research 10*, 2 (May), 180–184.

ITU-R. 1994. Methods for subjective assessment of small impairments in audio systems including multichannel sound systems, ITU-R BS 1116.

LAGRANGE, M., AND MARCHAND, S. 2001. Real-time additive synthesis of sound by taking advantage of psychoacoustics. In *Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-01), Limerick, Ireland, December 6-8*.

LARSSON, P., VÄSTFJÄLL, D., AND KLEINER, M. 2002. Better presence and performance in virtual environments by improved binaural sound rendering. *proceedings of the AES 22nd Intl. Conf. on virtual, synthetic and entertainment audio, Espoo, Finland* (June), 31–38.

LIKAS, A., VLASSIS, N., AND VERBEEK, J. 2003. The global *k*-means clustering algorithm. *Pattern Recognition 36*, 2, 451–461.

LOKKI, T., GRÖHN, M., SAVIOJA, L., AND TAKALA, T. 2000. A case study of auditory navigation in virtual acoustic environments. *Proceedings of Intl. Conf. on Auditory Display (ICAD2000)*.

MARTENS, W. 1987. Principal components analysis and resynthesis of spectral cues to perceived direction. In *Proc. Int. Computer Music Conf. (ICMC'87)*, 274–281.

MOORE, B. C. J., GLASBERG, B., AND BAER, T. 1997. A model for the prediction of thresholds, loudness and partial loudness. *J. of the Audio Engineering Society 45*, 4, 224–240. Software available at http://hearing.psychol.cam.ac.uk/Demos/demos.html.

MOORE, B. C. 1997. *An introduction to the psychology of hearing*. Academic Press, 4th edition.

PAINTER, E. M., AND SPANIAS, A. S. 1997. A review of algorithms for perceptual coding of digital audio signals. *DSP-97*.

PAQUETTE, E., POULIN, P., AND DRETTAKIS, G. 1998. A light hierarchy for fast rendering of scenes with many lights. *Proceedings of EUROGRAPHICS'98*.

PIERCE, A. 1984. *Acoustics. An introduction to its physical principles and applications*. 3rd edition, American Institute of Physics.

SAVIOJA, L., HUOPANIEMI, J., LOKKI, T., AND VÄÄNÄNEN, R. 1999. Creating interactive virtual acoustic environments. *J. of the Audio Engineering Society 47*, 9 (Sept.), 675–705.

SENSAURA, 2001. ZoomFX, MacroFX, Sensaura©. http://www.sensaura.co.uk.

SOUNDBLASTER, 2004. Creative Labs Soundblaster©. http://www.soundblaster.com.

STEIGLITZ, K. 1996. *A DSP Primer with applications to digital audio and computer music*. Addison Wesley.

TSINGOS, N., AND GASCUEL, J.-D. 1997. Soundtracks for computer animation: sound rendering in dynamic environments with occlusions. *Proceedings of Graphics Interface'97* (May), 9–16.

TSINGOS, N., FUNKHOUSER, T., NGAN, A., AND CARLBOM, I. 2001. Modeling acoustics in virtual environments using the uniform theory of diffraction. *ACM Computer Graphics, SIGGRAPH'01 Proceedings* (Aug.), 545–552.

VAN DEN DOEL, K., PAI, D. K., ADAM, T., KORTCHMAR, L., AND PICHORA-FULLER, K. 2002. Measurements of perceptual quality of contact sound models. In *Proceedings of the International Conference on Auditory Display (ICAD 2002), Kyoto, Japan*, 345–349.

VAN DEN DOEL, K., KNOTT, D., AND PAI, D. K. 2004. Interactive simulation of complex audio-visual scenes. *Presence: Teleoperators and Virtual Environments 13*, 1.

VROOMEN, J., AND DE GELDER, B. 2004. Perceptual effects of cross-modal stimulation: Ventriloquism and the freezing phenomenon. In *Handbook of multisensory processes*, G. Calvert, C. Spence, and B. E. Stein, Eds. M.I.T. Press.

WENZEL, E., MILLER, J., AND ABEL, J. 2000. A software-based system for interactive spatial sound synthesis. *Proceeding of ICAD 2000, Atlanta, USA* (April).

ZWICKER, E., AND FASTL, H. 1999. *Psychoacoustics: Facts and Models*. Springer. Second Upadated Edition.

PRIORITIZING SIGNALS FOR SELECTIVE REAL-TIME AUDIO PROCESSING

Emmanuel Gallo[1,2]*, Guillaume Lemaitre*[1] *and Nicolas Tsingos*[1]

[1]REVES-INRIA and [2] CSTB,
Sophia Antipolis, France.
Emmanuel.Gallo@sophia.inria.fr

ABSTRACT

This paper studies various priority metrics that can be used to progressively select sub-parts of a number of audio signals for real-time processing. In particular, five level-related metrics were examined: RMS level, A-weighted level, Zwicker and Moore loudness models and a masking threshold-based model. We conducted a pilot subjective evaluation study aimed at evaluating which metric would perform best at reconstructing mixtures of various types (speech, ambient and music) using only a budget amount of original audio data. Our results suggest that A-weighting performs the worst while results obtained with loudness metrics appear to depend on the type of signals. RMS level offers a good compromise for all cases. Our results also show that significant sub-parts of the original audio data can be omitted in most cases, without noticeable degradation in the generated mixtures, which validates the usability of our selective processing approach for real-time applications. In this context, we successfully implemented a prototype 3D audio rendering pipeline using our selective approach.

1. INTRODUCTION

Many applications ranging from video games to virtual reality or visualization/sonification require processing large number of audio signals in real-time. For instance, modern video games must render a large number of 3D sound sources using some form of spatial audio processing. Furthermore, each source's audio signal may itself be generated as a mixture of a number of sub-signals (e.g., a car-noise is a composite of engine and tire/surface noise) driven from real-time simulated physical parameters. The number of audio signals to process may often exceed hardware capabilities. Priority schemes that select the sounds to process according to a preset importance value are a common way of using hardware more efficiently, for instance by managing the limited number of hardware channels on a dedicated sound card. Usually, this value is determined by the sound designer at production time and might further be modulated by additional effects at run-time, such as attenuation of the sound due to distance or occlusion.

This paper is focused on the problem of automatically prioritizing audio signals according to an importance metric, in order to selectively process these signals. Such a metric can then be used to tune the processing "bit-rate" in order to fit a given computational budget: for instance, allocating a budget number of arithmetic operations to a complex signal processing task (e.g., a combination of mixing, filtering, etc.) involving a large number of source signals.

Figure 1 shows a basic example application where four speech signals have been prioritized according to a loudness metric and a mix has been generated simply by playing back the single most-important signal per processing frame (highlighted in yellow). Such an approach could typically be used for hardware voice management in video games.

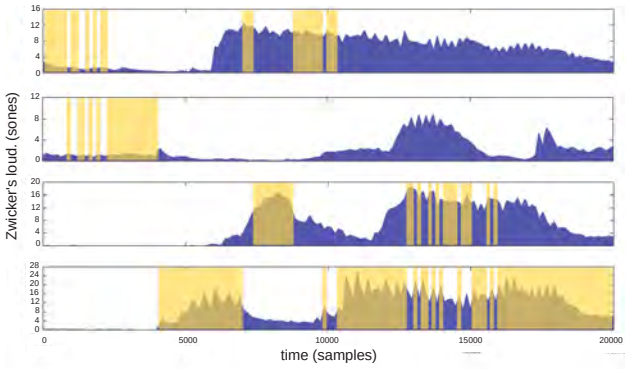


Figure 1: Four speech signals prioritized according to a loudness metric computed over successive short time-frames. The single most important frame across time is highlighted in yellow.

This paper presents a comparative study of several metrics that can be used to prioritize signals for selective real-time processing of audio signals. In section 2, we start by reviewing previous work related to scalable and progressive audio processing. A coarse-grain selective processing algorithm is described in section 3. In particular, several metrics that can be used to prioritize the audio signals and selectively allocate the required operations are discussed in section 3.1. Our selective processing algorithm is demonstrated in the context of a time-domain pipeline comprising mixing and simple filtering operations in section 3.2. Results of a pilot subjective study are presented in section 4 that support the applicability of our technique. We finally discuss our approach and outline other possible applications of our prioritization scheme before concluding.

2. RELATED WORK

While parametric, progressive and scalable codecs are a key research topic in the audio coding community [1, 2, 3], few attempts to date have been made to design scalable or selective approaches for real-time signal processing.

Fouad et al. [4] propose a level-of-detail rendering approach for spatialized audio where the sound samples are progressively generated based on a perceptual metric in order to respect a budgeted computing time. When it is elapsed, missing samples are interpolated from the calculated ones. As they prioritize signals

according to their overall energy, such a scheme will fail at capturing large energy variations through time within the signal itself.

Wand and Straßer [5] proposed a multi-resolution approach to 3D audio rendering. At each frame of their simulation, they use an importance sampling strategy to randomly select a sub-set of all sound sources to render. However, their importance sampling strategy also does not account for the variations in signal intensity. Such effects might be much more significant (factors of 10 or more can be easily observed on speech signals for instance) than variations in the control parameters such as distance attenuation, etc. since the latter usually vary smoothly and slowly through time (except for very near-field sources).

In a previous work [6], we proposed a framework for 3D audio rendering of complex virtual environments in which sound sources are first sorted by an *importance* metric, in our case the *loudness level* of the sound signals. We use pre-computed descriptors of the input audio signals (e.g., energy in several frequency bands through time) to efficiently re-evaluate the importance of each sound source according to its location relative to the listener. Hence, loudness variations within the signals are properly accounted for. The priority metric was used to determine inaudible sources in the environment due to auditory masking and group sound sources to optimize spatialization. This paper extends this approach by comparing several priority metrics and their subjective effect on selective processing of audio signals even for cases where removed sub-parts of the signals are above masking threshold.

Other scalable approaches based, for instance, on modal synthesis, have also been proposed for real-time rendering of multiple contact sounds in virtual environments [7, 8, 9]. Similar parametric audio representations [1, 2] also allow for scalable audio processing (e.g. pitch shifting or time-stretching, frequency content alteration, etc.) at limited additional processing cost, since only a limited number of parameters are processed rather than the full Pulse Code Modulation (PCM) audio data. However, this approach might imply real-time coding and decoding of the sound representations. As parametric representations are not widely standardized and commonly used in interactive applications, available standard hardware decoders do not usually give access to the coded representation in a convenient form for the user to further manipulate. Eventhough processing in coded domain might be achieved through modified software implementation of standard audio codecs (e.g. MPEG-1 layer 3, MPEG-2 AAC) [10], the overhead due to partial decoding would probably be overwhelming for a real-time application handling many signals.

3. SELECTIVE AUDIO PROCESSING

We propose a coarse-grain selective audio processing framework that can be separated into two steps : 1) we assign a priority to each frame of the input signals and 2) we select the frames to process by decreasing priority order until our pre-specified budget is reached. Remaining frames are simply discarded from the final result. Both steps are applied at each processing frame to produce a frame of processed output signal. The following sections detail both steps.

3.1. Priority metrics

In our approach, as well as others we described in section 2, processing management is driven by a given importance metric. The choice of this metric is then a crucial step: the audibility of the artefacts introduced by any processing optimizations will depend on its quality.

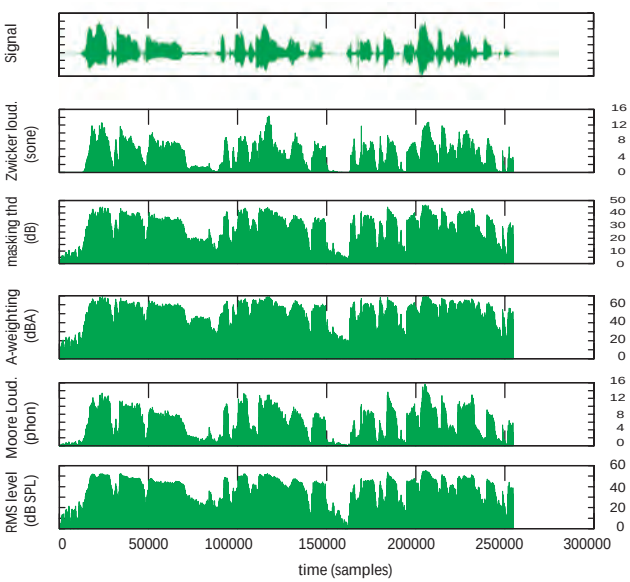


Figure 2: Several priority metrics calculated for an example speech signal using 3 ms-long frames.

Loudness seems a good candidate since it has been shown to be closely related to masking phenomena [11, 12]. Using loudness as an importance metric might hence allow important maskers to be processed first. But one can imagine that weighting may be more efficiently performed on the basis of more cognitive aspects. For instance, in the context of a collision avoidance experimental setup, Robert Graham [13] noticed that faster braking reaction times were measured when drivers were warned by car horns sounds, even if they were less loud than other tested sounds. There is a vast literature aiming at building psychoacoustic relationships between acoustic parameters of a sound and its so-called *urgency* (see Stanton and Edworthy [14] and [15] for an overview). Derivations of these urgency metrics may form a more cognitive-founded importance metric.

As a starting point, this paper examines the ability of several level-related metrics to optimize audio processing. In particular, we evaluated the following importance metrics:

1. RMS level, expressed in dB SPL,
2. *A*-weighted level, expressed in dBA [16],
3. Moore, Glasberg & Baer's loudness level [17], expressed in *phons*, calculated assuming a stimulus is a band-limited noise.
4. Zwicker's loudness [18], expressed in *sones*,
5. "Masking level" model defined as the level of the source minus a masking-threshold offset, expressed in relative dB (a masking threshold of -3 dB indicates that sounds with a energy weaker than half the energy of the masking sound will be masked), predicted from the *tonality* index of the signal [19, 20]. Tonality index is typically derived from a *spectral flatness measure* and indicates the tonal or noisy nature of the signal [21].

Each metric is evaluated for short processing frames along our test signals, typically every 3 to 23 ms (i.e., 128 to 1024 samples at 44.1kHz). Results were not significantly different for the various frame sizes. Smaller frames give better time-resolution and can result in more optimal interleaving of the signals during the processing step. However, frames too short can result in highly degraded audio information since interleaved signals will no longer be recognizable, a problem closely related to the illusion of continuity [22]. Figure 2 shows a comparison of several loudness metrics evaluated on a fragment of speech signal.

Table 1 shows the average rank correlation obtained with various metrics on three different mixtures of speech, ambient and music signals. Rank correlation measures how correlated the orderings obtained with the various metrics are. As can be seen in this table, results appear to be dependent on the type of signals. For speech and ambient sounds, metrics are correlated although not strongly. For the musical mixture, results are more pronounced showing stronger correlation between Zwicker's and Moore's loudness models and very low correlation between loudness models and all the others.

speech	Zwicker loud.	mask. thr.	Moore loud.	RMS level	A-weight.
Zwicker loud.	1 (0)	0.37 (0.22)	0.57 (0.29)	0.40 (0.22)	0.35(0.23)
mask thr.	0.37 (0.22)	1 (0)	0.54 (0.22)	0.73 (0.22)	0.56 (0.31)
Moore loud.	0.57 (0.29)	0.54 (0.22)	1 (0)	0.54 (0.23)	0.36 (0.21)
RMS level	0.40 (0.23)	0.73 (0.22)	0.54 (0.23)	1 (0)	0.54 (0.31)
A-weight.	0.35 (0.23)	0.56 (0.31)	0.36 (0.21)	0.54 (0.31)	1 (0)
ambient	Zwicker loud.	mask. thr.	Moore loud.	RMS level	A-weight.
Zwicker loud.	1 (0)	0.40 (0.18)	0.44 (0.18)	0.42 (0.19)	0.37 (0.17)
mask thr.	0.40 (0.18)	1 (0)	0.48 (0.17)	0.51 (0.18)	0.35 (0.18)
Moore loud.	0.44 (0.18)	0.48 (0.17)	1 (0)	0.47 (0.17)	0.37 (0.17)
RMS level	0.42 (0.19)	0.51 (0.17)	0.47 (0.17)	1 (0)	0.33 (0.18)
A-weight.	0.37 (0.17)	0.35 (0.17)	0.37 (0.17)	0.33 (0.18)	1 (0)
music	Zwicker loud.	mask. thr.	Moore loud.	RMS level	A-weight.
Zwicker loud.	1 (0)	0.05 (0.12)	0.42 (0.12)	0.04 (0.11)	0.03 (0.10)
mask thr.	0.05 (0.11)	1 (0)	0.04 (0.11)	0.42 (0.09)	0.40 (0.10)
Moore loud.	0.42 (0.12)	0.04 (0.11)	1 (0)	0.02 (0.10)	0 (0.10)
RMS level	0.04 (0.11)	0.42 (0.10)	0.02 (0.10)	1 (0)	0.36 (0.10)
A-weight.	0.03 (0.10)	0.40 (0.10)	0 (0.10)	0.36 (0.10)	1 (0)

Table 1: Rank correlation matrices for three test mixtures of speech, ambient and musical signals. Rank correlation was calculated using Spearman's formula [23] and averaged over all frames of the mixture. Its variance across frames is also given in brackets.

3.2. Selective processing algorithm

Our budget allocation algorithm is designed for real-time streaming applications. Hence, it has to be efficient and has to find a solution locally at each processing frame. To do this, the importance of each frame of the signal is evaluated and until our computational budget is reached, the algorithm selects which sub-parts of the signals should be processed, by decreasing priority value, using a greedy approach (i.e, taking the best immediate, or local, solution). An example is shown in Figure 1. The result is thus constructed as an interleaved mixture of the most important frames in all signals. To avoid artefacts during the reconstruction step, an overlap-add method (3ms frames with 10% overlap) was used. Another example is shown in Figure 3. Selected frames for different budgets are highlighted. As can be seen in the figure, our approach directly accounts for any sparseness in the mix by removing input frames below audibility threshold from the final mix. This might already result in a significant gain. For the various mixtures we used (ambient sounds, music and speech), we estimated that 0.7% to 33% of the input frames could be trivially removed (0.7% for ambient sounds, 24.5% for music and 33% for speech).

To improve the frequency resolution of our approach, we can further evaluate the priority metric for a number of sub-bands of the signals. In our experiments, we used four sub-bands corresponding to 0-500 Hz, 500-2000 Hz, 2000-8000 Hz, 8000-22000 Hz and treated each sub-band as if it were an additional input sound signal to prioritize. This would be typically useful for applications performing some kind of sub-band correction of the audio signal (e.g., equalizers). The required band-pass filtering can then be performed only on the selected sub-parts of the signal.

3.3. Integration within a real-time processing framework

Although most of the level-related priority metrics we used cannot be directly evaluated in real-time for large numbers of audio streams, they can be efficiently computed from additional descriptors stored with the audio data, in a manner similar to [6]. Loudness information, in particular, can be retrieved from precomputed loudness tables, energy levels and tonality indices stored with each corresponding frame of input audio data. This information may also be stored for several sub-bands of the signal. Such an approach allows us to further modulate the importance value in real-time depending on various other effects affecting the signal during the simulation. This necessary information is quite compact (typically about 1 to 4 Kb/sec of input audio data) and can be interleaved with standard PCM audio data for streaming or kept resident in memory for random access while the PCM data is streamed on-demand.

Our coarse-grain selective algorithm integrates well within standard time-domain audio processing pipelines. We evaluated it in the context of a 3D audio processing application for virtual reality. In this case, the signals of each virtual sound source undergo filtering and resampling operations to simulate propagation effects (atmospheric scattering, occlusion, Doppler shift, etc.) and binaural hearing (e.g., HRTF filtering) before being combined to produce the final mix. We implemented a scalable 3D audio processing pipeline implementing these effects using time-domain resampling and attenuation over several sub-bands of each source signal, computed using second-order biquad filters. Using our selective pipeline, we were able to process the signals using a budget number of operations resulting in a computing speed-up directly proportional to the selected budget. As sources of decreasing priority are processed, a complementary solution is to simplify the operations (for instance, using linear resampling instead of better quality spline-based resampling) rather than maintain high-quality processing for all selected frames and simply drop low priority frames. Example movie files demonstrating the approach are available at:
http://www-sop.inria.fr/reves/projects/scalableAudio/.

4. PILOT SUBJECTIVE EVALUATION

In order to evaluate subjective differences between the various metrics we ran a pilot evaluation study described in the following sections.

4.1. Experimental conditions

Subjects: 18 subjects (10 women and 8 men, 19 to 48 years old) volunteered as listeners. All reported normal hearing. Most of them were computer scientists, very few with any experience in acoustics or music practice. None of them was familiar with audio

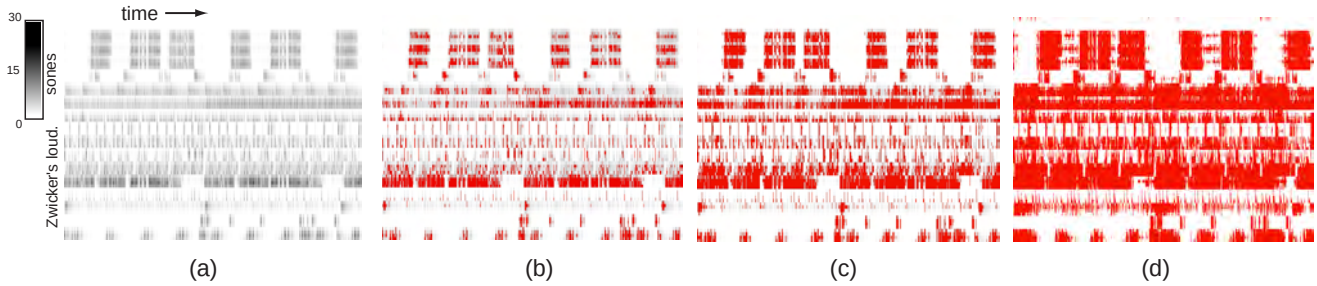


Figure 3: (a) Loudness values (using Zwicker's loudness model) through time for the 17 tracks of a musical mix. Each track was selectively filtered and processed into 4 frequency sub-bands resulting in 68 signals to prioritize. (b) Priority map showing the first 12.5% most important frames highlighted in red. (c) First 25% most important frames and (d) first 50% most important frames.

coding techniques, nor were regular users of mp3 or other coded-audio standards.

Stimuli: Three mixtures of various *types of signals* were generated: 1) a multi-track musical mix, 2) male and female Greek, French and Polish speech and 3) ambient sounds. The mixtures were created respectively from 17, 6 and 4 recordings separated into four sub-bands, resulting in 68, 24 and 16 signals to prioritize. Mixtures were generated at three *resolutions*, selecting the most important frames according to our priority metrics, using only 50%, 25% and 12.5% of the input signal data. Five different priority *metrics* (see section 3.1) were tested. A total of 45 stimuli (3 types of signals * 3 resolutions * 5 metrics) were hence created. All signals were presented at CD quality (44.1 kHz sampling rate and 16 bits quantization)[1].

Apparatus: We ran the test on a laptop computer using an in-house test program (see Figure 4). It was conducted using headphone presentation in a quiet office room. *Sennheiser HD600* headphones were used (diotic listening), calibrated to a reference listening level at eardrum (100 dB SPL). The sounds were stored on the computer hard drive and played through the SigmaTel C-Major integrated sound-board. They were played back at a comfortable level.

Procedure: The subjects were given written instructions explaining the task. They were asked to rate the 45 resulting output mixtures relative to the corresponding reference mix. We used the ITU-R[2] recommended *triple stimulus, double blind with hidden reference* technique, previously used for quality assessment of low bit-rate audio codecs [24]. Subjects were presented with three stimuli, R, A and B, corresponding to the reference, the test stimulus and a hidden reference stimulus (the hidden reference was the reference itself, without any alteration)[3]. Test stimuli were presented to each subject in a different (random) order. The hidden reference was randomly assigned to button A or B. Our test program automatically kept track in an output log file of the presentation order and the marks given respectively to the stimulus and the hidden reference signal for each test. The output of the procedure was then two scores: one for the hidden reference, and one the for the test stimulus. After the ITU-R standard, the judgment value used for further analysis was the difference between the scores of the hidden reference and test stimulus. Hence, a value of zero indicates than no difference was perceived between the reference and the test sound. A positive score indicates that an annoying difference was heard between the test sound and the reference sound, and a negative score indicates that the test sound was better rated than the reference sound.

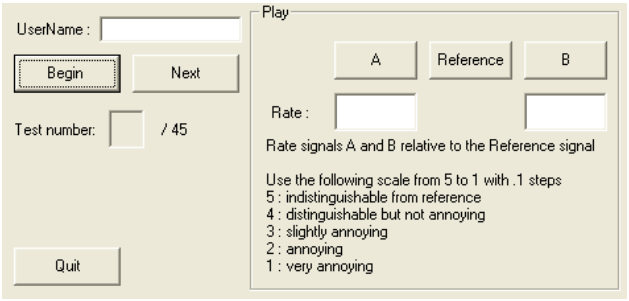


Figure 4: Snapshot of the interface designed for our listening tests.

Subjects could switch between the three stimuli at any time during playback by pressing the corresponding buttons on the interface (see Figure 4). They were asked to rate differences between each test stimuli (A and B) and the Reference from "imperceptible" to "very annoying", using a scale ranging from 5.0 to 1.0 (with one decimal) [25].

After the test, subjects were invited, during a semi-guided interview, to describe the differences that they heard between the processed and the original sounds.

4.2. Analysis

Correlations between the subjects: All subjects raw judgments were significantly correlated ($p < 0.01$) except for one who was removed from further analysis. After removing this subject, the correlation coefficients ranged from 0.38 to 0.92.

Analysis of variance: A three-way analysis of variance was performed over the judgments (repeated design). Results are given in Table 2. The experimental factors affecting the judgments are: S: subjects, R: resolution , M: metric and T: type of signals. All principal effects are significant (resolution: F(2,32)=195.0, p corrected < 0.01; metric: F(4,64)=16.3, p corrected < 0.01; type of signal: F(2,32)=41.1, p corrected < 0.00). Only the interactions between between *metric* and *type of signals* is significant at the lower threshold (F(8,128)=8.6 p corrected < 0.01). The principal effects of the experimental factors are depicted in Figure 5

[1] The stimuli used for the tests can be found at:
`http://www-sop.inria.fr/reves/projects/scalableAudio/`

[2] International Telecommunication Union

[3] *i.e.*, the subjects did not know which of A or B was the actual test or the reference.

Source	df	Sum of squares	Mean squares	F-value	p cor.
S	16	139.8	8.7		
R	2	967.7	483.2	195.0	0.000(**)
*S*R*	32	79.4	2.5		
M	4	23.4	5.8	16.3	0.001(**)
*S*M*	64	23.0	0.3		
*R*M*	8	5.6	0.7	1.9	0.191 (ns)
*S*R*M*	128	48.2	0.4		
T	2	112.7	56.3	41.1	0.000 (**)
*S*T*	32	43.8	1.4		
*R*T*	4	27.5	6.9	6.8	0.0191(*)
*S*R*T*	64	64.7	1.0		
*M*T*	8	24.5	3.0	8.6	0.010(**)
*S*M*T*	128	45.5	0.4		
*R*M *T*	16	17.2	1.07	2.8	0.115(ns)
*S*R*M*T*	256	99.1	0.4		
Total		1722.0	2.2		

df: degree of freedom
p cor.: corrected probability (conservative F-test)
* $p<0.05$; ** $p<0.01$; ns: not significant

Table 2: Anova table for the subjective evaluation

(vertical bars represent the standard deviation).

The bottom graph in this figure clearly shows the effect of the resolution on the average judgments: when 50% of the data are kept, average judgments lay between 0 and 1, almost meeting the requirement for transparency (i.e., no judgment above 1). At 25% resolution, average judgments rise to values between 1 and 2.5, and slide up to more than 3.5 for a resolution of 12.5%. The top graph in the figure indicates that musical signals were, on average, better ranked than the other type of signals (subjects freely mentioned during post-experimental interviews that differences were harder to notice for musical sounds). This indicates that the alterations of the signal produced by the algorithm are less perceptible for musical sounds. The middle graph in the figure represents the effect of the metric on the average judgments. Results were not quite as pronounced, but a first conclusion is that the A-weighting metric leads to the most audible difference between the processed and original sounds. Further understanding is obtained by studying the significant interaction between metric and type of signal, depicted in Figure 6.

The patterns of effects for the metrics are qualitatively identical for both musical and speech signals: A-weighting leads to the worst results, RMS level and Zwicker's loudness model result in the best judgments, Moore's loudness yields to slightly weaker judgments. On the other hand, for ambient sounds, Zwicker's loudness model results in the worst judgments, whereas Moore's loudness model leads the processed sounds to be better evaluated with reference to the original ones. Although, our evaluation was aimed at a totally different purpose, our results share some similarities with the recent paper by Skovenborg and Nielsen [26] that classified twelve loudness models (including several RMS-level metrics, Zwicker loudness and A-weighted level) into four categories. In their experiments, A-weighting was found to perform worst as a loudness metric while no clear advantage was found

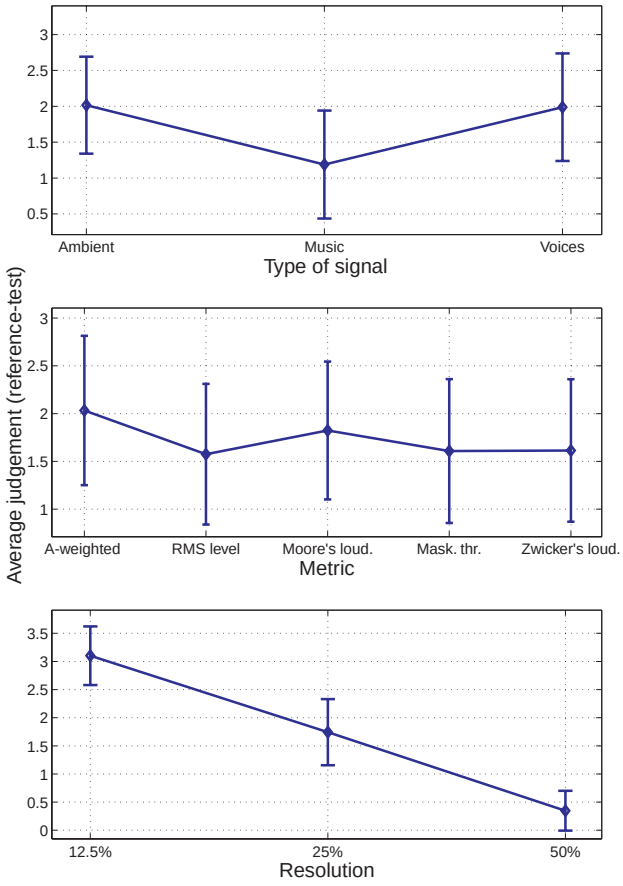


Figure 5: Principal effects of the experimental factors. Vertical bars represent standard deviation. The average judgment represents an annoyance level (hidden reference minus stimulus).

for the Zwicker loudness model over RMS-level related metrics. However, we could not test their two new loudness models, which seem to perform best. This would be an interesting future study to conduct.

5. DISCUSSION

Several conclusions can be drawn from this study. First of all, when only 50 % of the original data are used, subjects are almost unable to hear any difference between processed and original mixtures. When only 25 % of the sounds are preserved, average judgments lay between 2.5 and 1 (respectively "slightly annoying" and "perceptible but not annoying"). This indicates that our algorithm can reduce the required number of operations by more than 50 % without dramatically distorting the resulting mixtures (see Figure 7).

Another conclusion is that the judgments seem to be strongly influenced by the type of signal. However, as this variable also integrates several other effects (numbers of signals in the mix, sparseness of the mix, energy distribution in the mix, etc.) it would require further testing.

Nevertheless, differences between original and processed sounds were more difficult to detect for musical sounds. Two hypothesis

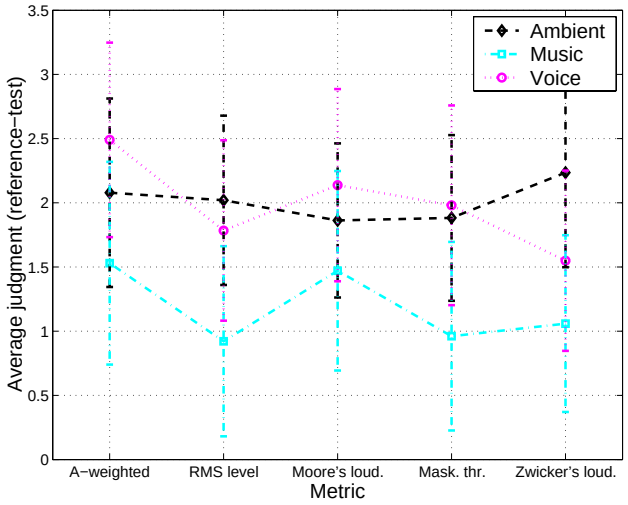


Figure 6: Interactions between the effects of the metric and the type of signal on the average judgments. The average judgment represents an annoyance level (hidden reference minus stimulus).

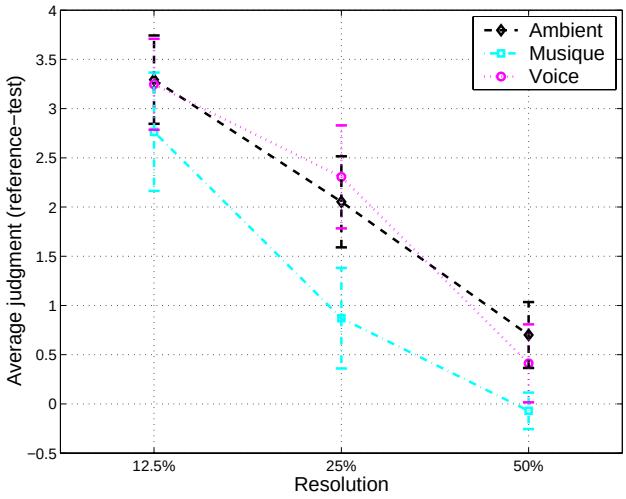


Figure 7: Averaged judgments for the three test mixtures and for three levels of detail. When only 50% of the input audio data was used, the resulting mixture was highly rated regardless of the stimuli.

can be formulated to explain this phenomenon: first of all, due to their nature, musical sounds are more sparse than other sounds. Energy peaks occur at regular rhythmic patterns, and there might a significant amount of low energy frames between these rhythmic accents. In our example, we estimated the sparseness (ratio between silence and signal) of our musical mix to be about 25%, which would make it well suited to our algorithm. However, the speech mixture was found to be much sparser that the ambient mixture (33% vs. less than 1%) although the results for these two cases were rather similar.

Another hypothesis is that the metrics were, in general, better suited to musical sounds.

Comparing loudness models, Zwicker's model leads to better results for speech and musical sounds, while Moore's loudness model performs best for ambient sounds. This is consistent with our implementation of Moore's loudness model for noisy signals (it can be reasonably assumed that ambient sounds are noisier than musical sounds).

These conclusions were confirmed during the interviews of the subjects. Many subjects reported that they used different criteria for the different types of signals. For speech signals, they reported to produce favorable judgments as long as the intelligibility was preserved, although most of the mixture was foreign language to them. For musical sounds, they did not hear any difference until the sounds were drastically distorted. Finally, for ambient sounds, they seem to have performed some kind of "spectral listening"; a typical remark being: "I tried to notice if there was more or less bass/treble". Hence, we can conclude that no metric seems to perform best in all cases but, rather, that the importance metric has to be adapted to the type of signal.

6. CONCLUSIONS

We have presented an approach for coarse-grain selective processing of audio signals. Several level-related metrics that can be used to drive the selection process were compared showing significant difference between the various metrics in terms of the ordering induced on the signals. A pilot subjective evaluation study suggests that A-weighting does not perform as well as the other metrics at prioritizing the sound signals. While RMS level appears as a good compromise, other metrics, loudness in particular, can yield to better results depending on the type of signals. Our selective processing approach integrates well within standard audio processing pipelines and can be used to reduce the necessary operations by 50% while remaining near-transparent or 75% with an acceptable degradation of the perceived quality.

As future work, we would like to explore extensions to finer-grain processing by combining our selection scheme with parametric audio coders or alternate representations for audio signals.

We believe that proposing and evaluating more sophisticated priority metrics is of primary interest for a wide range of applications including memory/resource management (e.g. 3D hardware voices, streaming from main storage space), real-time masking evaluation [6], on-the-fly multi-track mixing [27], dynamic coding and transmission of spatial audio content and more generally for computational auditory scene analysis.

7. ACKNOWLEDGEMENTS

The authors would like to thank the anonymous reviewers for their useful comments. This research was supported in part by the 2-year RNTL project OPERA, co-funded by the French Ministry of Research and Ministry of Industry.
http://www-sop.inria.fr/reves/OPERA.

8. REFERENCES

[1] B. L. Vercoe, W. G. Gardner, and E. D. Scheirer, "Structured audio: Creation, transmission, and rendering of parametric sound representations," in *Proceedings of IEEE*, may 1998, vol. 86, pp. 922–939.

[2] H. Purnhagen, "Advances in parametric audio coding," in

IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA'99), New-Paltz, NY, 1999.

[3] J. Herre, "Audio coding - an all-round entertainment technology," in *Audio Engineering Society 22nd International Conference on Virtual, Synthetic and Entertainment Audio (AES'22), Espoo, Finland*, June 15-17 2002, pp. 139–148.

[4] H. Fouad, J.K. Hahn, and J.A. Ballas, "Perceptually based scheduling algorithms for real-time synthesis of complex sonic environments," *proceedings of the 1997 International Conference on Auditory Display (ICAD'97), Xerox Palo Alto Research Center, Palo Alto, USA*, 1997.

[5] W. Straßer and M. Wand, "Multi-resolution sound rendering," in *Symp. Point-Based Graphics*, 2004.

[6] N. Tsingos, E. Gallo, and G. Drettakis, "Perceptual audio rendering of complex virtual environments," *ACM Transactions on Graphics (Proceedings of SIGGRAPH'04)*, vol. 23, no. 3, 2004.

[7] M. Lagrange and S. Marchand, "Real-time additive synthesis of sound by taking advantage of psychoacoustics," in *Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-01), Limerick, Ireland, December 6-8*, 2001.

[8] K. van den Doel, D. K. Pai, T. Adam, L. Kortchmar, and K. Pichora-Fuller, "Measurements of perceptual quality of contact sound models," in *Proceedings of the International Conference on Auditory Display (ICAD 2002), Kyoto, Japan*, 2002, pp. 345–349.

[9] K. van den Doel, D. Knott, and D. K. Pai, "Interactive simulation of complex audio-visual scenes," *Presence: Teleoperators and Virtual Environments*, vol. 13, no. 1, 2004.

[10] A. B. Touimi, "A generic framework for filtering in subband-domain," in *Proc. of the Ninth DSP Workshop (DSP2000), Hunt, Texas*, 2000.

[11] E. Zwicker, "Dependence of post-masking on masker duration and its relation to temporal effects in loudness.," *Journal of the Acoustical Society of America*, vol. 75, no. 1, pp. 219–223, Jan 1984.

[12] F. Baumgarte, "A physiological ear model for auditory masking applicable to perceptual coding," in *Proc. of 103rd Convention of the Audio Engineering Society, , New York, USA*, 1997.

[13] R. Graham, "Use of auditory icons as emergency warnings: evaluation within a vehicle collision avoidance application," *Ergonomics*, vol. 42, no. 9, pp. 1233–1248, 1999.

[14] N. A. Stanton and J. Edworthy, "Auditory warnings and displays: an overview," in *Human Factors in Auditory Warnings*. Ashgate Publishing Ltd., 1999.

[15] N. A. Stanton and J. Edworthy, Eds., *Human Factors in Auditory Warnings*, Ashgate Publishing Ltd., 1999.

[16] Acoustics FAQ Section 8.1, "A-weighting formula," http://www.faqs.org/faqs/physics-faq/acoustics/.

[17] B. C. J. Moore, B. Glasberg, and T. Baer, "A model for the prediction of thresholds, loudness and partial loudness," *J. of the Audio Engineering Society*, vol. 45, no. 4, pp. 224–240, 1997, Software available at http://hearing.psychol.cam.ac.uk/Demos/demos.html.

[18] E. Zwicker, H. Fastl, U. Widmann, K. Kurakata, S. Kuwano, and S. Namba, "Program for calculating loudness according to din 45631 (iso 532b)," *Journal of the Acoustical Society of Japan*, vol. 12, pp. 39–42, 1991.

[19] T. Painter and A. Spanias, "A review of algorithms for perceptual coding of digital audio signals," in *Proceedings of the International Conference on Digital Signal Processing*, 1997, pp. 179–205.

[20] K. Brandenburg, "mp3 and AAC explained," *AES 17th International Conference on High-Quality Audio Coding*, Sept. 1999.

[21] E. Zwicker and H. Fastl, *Psychoacoustics: Facts and Models*, Springer, 1999, Second Upadated Edition.

[22] S. McAdams, M.-C. Botte, and C. Drake, "Auditory continuity and loudness computation," *Journal of the Acoustical Society of America*, vol. 103, no. 3, pp. 1580–1591, March 1998.

[23] D. C. Howell, *Statistical methods for psychology*, PWS-Kent, 1992.

[24] C. Grewin, "Methods for quality assessment of low bit-rate audio codecs," *proceedings of the 12th AES conference*, pp. 97–107, 1993.

[25] ITU-R, "Methods for subjective assessment of small impairments in audio systems including multichannel sound systems, ITU-R BS 1116," 1994.

[26] E. Skovenborg and S. Nielsen, "Evaluation of different loudness models with music and speech material," in *Proc. of 117th Convention of the Audio Engineering Society, San Francisco*, 2004.

[27] F. Pachet and O. Delerue, "On-the-fly multi track mixing," in *Proceedings of the 109th Audio Engineering Society Convention*, 2000.

SCALABLE PERCEPTUAL MIXING AND FILTERING OF AUDIO SIGNALS USING AN AUGMENTED SPECTRAL REPRESENTATION

Nicolas Tsingos

REVES - INRIA
Sophia Antipolis, France
nicolas.tsingos@sophia.inria.fr

ABSTRACT

Many interactive applications, such as video games, require processing a large number of sound signals in real-time. This paper proposes a novel perceptually-based and scalable approach for efficiently filtering and mixing a large number of audio signals. Key to its efficiency is a pre-computed Fourier frequency-domain representation augmented with additional descriptors. The descriptors can be used during the real-time processing to estimate which signals are not going to contribute to the final mixture. Besides, we also propose an importance sampling strategy allowing to tune the processing load relative to the quality of the output. We demonstrate our approach for a variety of applications including equalization and mixing, reverberation processing and spatialization. It can also be used to optimize audio data streaming or decompression. By reducing the number of operations and limiting bus traffic, our approach yields a 3 to 15-fold improvement in overall processing rate compared to brute-force techniques, with minimal degradation of the output.

1. INTRODUCTION

Many interactive applications such as video games, simulators and visualization/sonification interfaces require processing a large number of input sound signals in real-time (e.g., for spatialization). Typical processing includes sound equalization, filtering and mixing and is usually performed for each of the inputs individually. In modern video games, for instance, hundreds of audio samples and streams might have to be combined to re-create the various spatialized sound effects and background ambiance. This results in both a large number of arithmetic operations and heavy bus traffic. Although consumer-grade audio hardware can be used to accelerate some pre-defined effects, the limited number of simultaneous hardware voices calls for more sophisticated voice-management techniques. Besides, contrary to their modern graphics counterparts, consumer audio hardware accelerators still implement fixed-function pipelines which might eventually limit the creativity of audio designers and programmers. Hence, designing efficient software solutions is still of major interest.

While perceptual issues have been a key aspect in the field of audio compression (e.g., mp3) [1, 2], most software audio processing pipelines still use brute-force approaches which are completely independent of the signal content. As a result, the number of audio streams they can process is usually limited rapidly since the amount of processing cannot be adapted on-demand to satisfy a predefined time vs. quality tradeoff. This is especially true for multi-media or multi-modal applications where only a small fraction of the CPU-time can be devoted to audio processing.

In recent years, several contributions have been introduced that aim to bridge the gap between perceptual audio coding and audio processing in order to make audio signal processing pipelines more efficient. A family of approaches proposed to directly process perceptually-coded audio signals [3, 4, 5, 6, 7] yielding faster implementations than a full decode-process-recode cycle. Although they are well suited to distributed applications involving streaming over low-bandwith channels, they require specific coding of the filters and processing. Moreover, they cannot guarantee an efficient processing for a mixture of several signals, nor that they would produce an "optimal" processing for the mixture.

Others, inspired by psycho-acoustic research and audio coding work, tried to use perceptual knowledge to optimize various applications. For instance, a recent paper by Tsingos et al. [8] proposed a real-time voice management technique for 3D audio applications which evaluates audible sound sources at each frame of the simulation and groups them into clusters that can be directly mapped to hardware voices. Necessary sub-mixing of all sources in each cluster is done in software at fixed-cost. Dynamic auditory masking estimation has also been successfully used to accelerate modal synthesis [9, 10]. In the context of long FIR filtering for reverberation processing, the recent work by Lee et al. [11] also shows that significant improvement can be obtained by estimating whether the result of the convolution is below hearing threshold, hence reducing the processing cost. In this paper, we build on these approaches and propose a scalable, perceptually-based, audio processing strategy that can be applied to a frequency-domain processing pipeline performing filtering and mix-down operations on a large number of input audio signals [12]. Key to our approach is the choice of a signal representation that allows its progressive encoding and reconstruction. In this paper, we use Fourier-transform domain as a convenient and widely used solution which satisfies these constraints. In this context, we present a set of techniques to:

- dynamically maintain features of the input audio signals to process (for instance, based on pre-computed information on the input audio samples in a way similar to [8]),
- dynamically evaluate auditory masking between a number of input audio frames that have to be processed and mixed-down to produce a frame of audio output,
- implement a scalable processing pipeline by fitting a pre-defined budget of operations to the overall task based on the importance of each input audio signal.

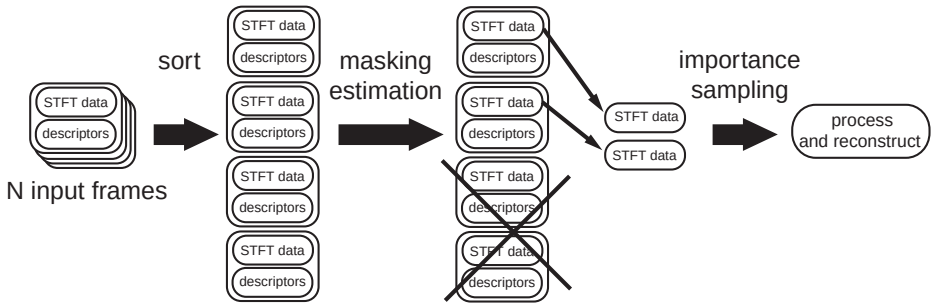


Figure 1: Overview of our scalable perceptual pipeline. All input signals frames at time t are first sorted according to their energy content and a masking estimation is performed. Audible frames are then sampled according to an importance metric so that only a subpart of their pre-computed STFT coefficients are processed to produce the output frame at t.

2. OVERVIEW OF OUR APPROACH

Our approach can be decomposed into four main stages (see Figure 1). The first stage builds a frequency domain representation of the audio signals based on a short-time Fourier transform (STFT). This representation is augmented by a set audio descriptors such as the root-mean-square (RMS) level of the signal in several frequency bands and the tonality [1] of the signal. This kind of augmented description of audio signals is also similar in spirit to prior work in indexing and retrieval of audio [13]. This first stage is usually performed off-line when the signals to process are known in advance.

The three remaining stages: masking evaluation, importance sampling and actual processing are performed on-line during the interactive application. Audio signals are processed using small frames of audio data (typically using windows of 20 to 40 ms) and, as a consequence, all three later steps are performed for each frame of the computed output stream.

The masking step determines which subset of the input audio frames will be audible in the final mixture. It is not mandatory but usually makes the importance sampling step more efficient. It can also be used to limit bus traffic since all inaudible signals can be directly discarded after the masking evaluation and do not have to go through the actual processing pipeline.

The importance sampling step determines the amount of data to select and process in each input signal in order to fit the predefined operation budget and minimize audible degradations. Finally, the actual processing step performs a variety of operations on the audio data prior to the final mix-down.

In the remainder of the paper, sections 3 to 5 detail these four stages while section 6 presents example applications of our techniques in the context of equalization/mixing, reverberation processing and 3D audio rendering.

3. PRE-PROCESSING AUDIO SIGNALS

The first stage of our approach aims at pre-computing a signal representation from which the later real-time operations can be efficiently performed. We chose a representation based on a STFT of the input signals augmented with additional information.

3.1. Constructing the representation

For each frame of the input audio signal, we first compute the STFT of the audio data. For 44.1 kHz signals, we use 1024 sample Hanning-windowed frames with 50% overlap, resulting in 512 complex values in frequency domain. From the complex STFT, we then compute a number of additional descriptors:

- RMS level for a predefined set of i frequency bands (e.g., octave or Bark bands),
- Tonality T calculated as a spectral flatness measure [1]; tonality is a descriptor in $[0, 1]$ encoding the tonal (when close to 1) or noisy (when close to 0) nature of the signal,
- Reconstruction error indicator Err; this descriptor indicates how well the signal can be reconstructed from a small number of bins.

To compute the indicator Err, we first sort the FFT bins by decreasing modulus value. Then, several reconstructions (i.e., inverse Fourier transforms) are performed using an increasing number of FFT bins. The reconstruction error, calculated as the RMS level of the (time-domain) difference between the original and reconstructed frame, is then computed. For a N bin FFT, we perform k reconstructions using 1 to N FFT bins, in N/k increments. Err is calculated as the average of the k corresponding errors values. This indicator will be later used during the on-line importance sampling step.

Descriptors, together with the pre-sorted FFT bins, are computed for each frame of each input signal and pre-stored in a custom file-format. If required, descriptors can be stored separately from the FFT data used for the processing. They can be viewed as a compact representation of the signal, typically requiring a few additional kBytes of data per second of audio signals (e.g., 3kBytes/sec. at 44.1kHz for 1024 sample frames with 50% overlap and 8 frequency bands). Hence, for a set of short audio signals, they could easily fit into memory for fast random access over all signals.

3.2. Optimizing the representation

This representation can be further optimized if necessary during the pre-processing step. Frames whose energy is below audible threshold can be stored with a minimal amount of data. Basic masking calculations can also be performed while computing Err by examining the signal-to-noise ratio between the energy in the selected FFT bins and the resulting reconstruction error. The number of stored bins can then be limited as soon as the signal-to-noise exceeds a specified threshold, which can further depend on the tonality of the signal [1]. Of course, any optimization made at this stage would imply that the signals are not going to be drastically

modified during the processing step. However, this restriction applies to any approach applied to audio data encoded using a lossy audio compression strategy.

Although compression is not the primary goal of our paper, we also experimented with various strategies to optimize storage space. By quantizing the complex FFT data with non-uniform 16-bit dynamic range and compressing all the data for each frame using standard compression techniques (e.g., zip), the size of the obtained sound files typically varies between 1.5 times (for wide-band sounds) and 0.25 times the size (e.g. speech) of the original 16-bit PCM audio data. If more dynamic range is necessary, it is also possible to quantize the n first FFT bins, which contain most of the energy, over a 24-bit dynamic range and to represent the rest of the data using a more limited range with minimal impact on the size of the representation and quality of the reconstruction.

4. REAL-TIME MASKING EVALUATION

Once the input sound signals have been pre-processed, we can use the resulting information to optimize a real-time pipeline running during an interactive application.

The first step of our pipeline aims at evaluating which of the input signals are going to significantly contribute to a given frame of the output, which amounts to evaluating which input signals are going to be audible in the final mixture at a given time. Signals that have been identified as inaudible can be safely removed from the pipeline reducing both the arithmetic operations to perform and the bus traffic. Since the calculation must be carried on at each processing frame, it must be very efficient so that it does not result in significant overhead. The masking algorithm is similar to the one presented in [8] and makes use of the the pre-computed descriptors (see Section 3.1) for maximum efficiency.

First, all input frames are sorted according to some importance metric. In [8], a loudness metric was used but some of our recent experiments seem to indicate that the RMS level would perform equally well, if not better on average, for lack of a "ultimate" loudness metric [14]. If the signals must undergo filtering or equalization operations, we dynamically weight the RMS level values pre-computed for several frequency-bands to account for the influence of the filtering operations in each band. We can then compute the importance as the sum of all weighted RMS values.

Second, all signals are considered in decreasing importance order for addition to the final mixture according to the following pseudo-code:

$\mathbf{M_{mix}} = 200$
$\mathbf{P_{mix}} = 0$
$\mathbf{T} = 0$
$\mathbf{P_{toGo}} = \sum_k \mathbf{RMS_k}$
while $(dB(\mathbf{P_{toGo}}) > dB(\mathbf{P_{mix}}) - \mathbf{M}_{mix})$
and $(\mathbf{P_{toGo}} > \mathbf{ATH})$ **do**
tag signal S_k as audible
$\mathbf{P_{toGo}} -= \mathbf{RMS_k}$
$\mathbf{P_{mix}} += \mathbf{RMS_k}$
$\mathbf{T} += \mathbf{P_k} * \mathbf{T_k}$
$\mathbf{T_{mix}} = \mathbf{T}/\mathbf{P_{mix}}$
$\mathbf{M_{mix}} = 27 * \mathbf{T_{mix}} + 6 * (1 - \mathbf{T_{mix}})$
k++
end

This process basically adds the level RMS_k of each source to an estimate of the level of the final result in each band P_{mix} (initially set to zero). Accordingly, it subtracts it from an estimate of the remaining level in each band P_{toGo} (initially set to the sum of all RMS levels for all signals). The process stops when the estimated remaining level in each band is below a given threshold M_{mix} from the estimated level of the final result. The process also stops if the remaining level is below the absolute threshold of hearing ATH [15]. Threshold M_{mix} is adjusted according to the estimated tonality of the final result T_{mix}, following rules similar to the ones used in perceptual audio coding [1]. In our applications, a simple constant threshold of -27 dB also gave satisfying results indicating that pre-computing and estimating tonality values is not mandatory. Note that all operations must be performed for each frequency band, although we simplified the given pseudo-code for the sake of clarity (accordingly, all quantities should be interpreted as vectors whose dimension is the number of used frequency bands and all arithmetic operations as vector arithmetic). In particular the process stops when the masking threshold is reached for *all* frequency bands.

5. IMPORTANCE SAMPLING AND PROCESSING

The second step of our pipeline aims at processing the sub-set of audible input signals in a scalable manner while preserving the perceived audio quality. In our case this is achieved by performing the required signal processing using a target number of operations over a limited sub-set of the original signal data. Note that it is not mandatory to perform masking calculations (as described in the previous section) in order to implement the following importance sampling scheme. However, the masking step limits the number of samples going through the rest of the pipeline and ensures that no samples will be wasted since our sampling strategy itself does not ensure that masked signals will receive a zero-sample budget.

5.1. Selecting a budget number of FFT bins

We can assume a constant number of arithmetic operations will be required for each complex FFT coefficient (i.e., bin). Hence, fitting a budget number of operations for our pipeline at each processing frame directly amounts to selecting a budget number of FFT bins for each frame of input sound signal. We can take advantage of pre-storing our FFT in decreasing energy order by directly processing the n_i first FFT bins for each input signal s_i so that $\sum_i n_i$ does not exceed our budget N.

We select n_i as being directly proportional to an importance value I_i calculated for each audio signal. In our case, we define I_i as:

$$I_i = log(1 + E_i * (1 + Err_i)), \quad (1)$$

where Err_i is the error indicator of signal s_i and E_i is the summed RMS level of the signal in all bands (including possible effects of filtering). As for the masking calculation, we use the RMS level value as the primary importance value. We further weight this value according to the error indicator of the signal, so that signals requiring more FFT bins to achieve a good reconstruction get higher priority.

The importance value I_i is then normalized so that $\sum_i I_i = 1$ and the number of bins to process for each signal is simply obtained as: $n_i = N\ I_i$. Note that the overall target number is an upper bound and might not be exactly met. For instance, when an optimized input representation is used, some frames might contain

a number of FFT bins smaller than their calculated n_i. Currently, we do not re-affect the additional number of bins to another signal.

5.2. Processing and reconstruction

Once the most-important FFT bins have been selected for all signals according to our target budget, they are simply sent to the actual processing pipeline. All calculations are done in frequency-domain so that a single inverse FFT is required to obtain the final time-domain audio signal. Since we pre-compute FFT data for 50% overlapped frames, reconstructing a time-domain frame for playback requires processing two frequency-domain frames, involving two inverse FFTs and an overlap-add operation.

6. APPLICATIONS AND RESULTS

We implemented and tested our scalable processing algorithm for three applications: a simple mixing and equalization pipeline, an FIR reverberation pipeline and a massive spatial audio application which can render hundreds of simultaneous sound sources in real-time. Example results are available for listening at the following URL: www-sop.inria.fr/reves/projects/dafx05. All examples were implemented in *C++* without any specific optimization and tested on a standard laptop computer with a *Pentium 4m* 1.8 GHz processor. All processing was done using 32-bit floating point arithmetic. Sampling rate was 44.1 kHz and we used output frames of 1024 samples. Hence, the output rate for audio data was 43 Hz. Masking calculations were performed using 15 frequency subbands.

6.1. Mixing and equalization

Our first application performs simple equalization and mix-down operations on a number of input sound signals. In this case, all data was streamed and decompressed from the disk in real-time. Even for a relatively small number of signals to process (in our test example we used 8), our approach shows a 3-fold speed-up compared to processing the entire data set. Overall processing speed, including streaming from disk and final time-domain reconstruction, is doubled. Table 1 shows a compute-time breakdown for various stages of the pipeline when processing the STFT data at several "resolutions", decreasing the number of target FFT bins. The results, expressed in Hz, correspond to the rate at which a full frame of output can be calculated. The load rate corresponds to the rate at which the audio data can be streamed from disk. For the load rate and processing rate, the figures are given per input signal. The total rate is given for the entire process applied to all signals, including streaming from disk. In this case, the total rate is streaming-bound. Hence, using an optimized representation, as described in section 3.2, brings a more than 2-fold improvement (as can be seen in the last column). In this case, near-transparent processing could be achieved in 1/10th of the duration of an output frame.

6.2. Block FIR filtering and reverberation

Our second application targets long FIR filters as used for reverberation processing. A common technique to implement low latency convolution with long filters is to decompose the filter in successive smaller blocks. These blocks can be of constant length [16, 17] or can be adapted to optimize the number of arithmetic operations [18]. Our technique extends the recent approach by Lee et al. [11] by 1) determining which part of the reverberation filter will not be audible due to auditory masking and 2) allowing for scalable rendering of the reverberation. In fact, this application is very similar to our previous example: each block of the reverberation filter can be convolved with separate delayed copies of the original signal (by multiples of one frame) and the results are mixed together to produce one frame of reverberant output. We tested our perceptual reverberation algorithm with artificial reverberation FIR filters created from exponentially decaying white noise [19]. However, our approach could be applied to any measured impulse response. Table 2 shows results of our approach applied to a stereo rendering of up to 12-second long reverberation filters (corresponding to about 1000 blocks of 512 time-samples). In the corresponding examples, the exponential decay was chosen to obtain a reverberation time of about 4.5 seconds. Here again, we show a compute-time breakdown for various stages of the pipeline when processing the STFT data at several "resolutions", decreasing the number of target FFT bins. The two left-most columns, denoted "all" and "all (mask)" correspond respectively to full processing (i.e., reference) and processing of all FFT bins of all *audible* audio frames (i.e., masking is turned on but all audible data is processed).

target FFT bins	all (4096)	2000	500	500 (optim.)
load rate (Hz)	1900	1896	1920	4500
processing rate (kHz)	60	116	180	200
total rate (Hz)	207	209	213	440
avg. masked frames (%)	0	15	15	15

Table 1: Breakdown performance figures for our test mixing and equalization pipeline. Load and processing rate are given per signal. All values are time-averaged over 29 seconds of processing-time.

Rotor example					
target FFT bins	all	all (mask)	50000	10000	5000
masking/sampling rate (Hz)	4754	2322	2263	2340	2387
processing rate (Hz)	82	263	809	1965	3108
FFT rate (Hz)	2109	2068	2124	2120	2110
total rate (Hz)	38	105	230	350	405
avg. masked frames (%)	0	87	87	87	87
Song example					
target FFT bins	all	all (mask)	50000	10000	5000
masking/sampling rate (Hz)	2452	1282	1273	1306	1316
processing rate (Hz)	40	165	570	1666	2556
FFT rate (Hz)	2126	2107	2084	2144	2129
total rate (Hz)	19	68	164	270	304
avg. masked frames (%)	0	80	80	80	80
Voice example					
target FFT bins	all	all (mask)	50000	10000	5000
masking/sampling rate (Hz)	13632	6387	6172	6004	6670
processing rate (Hz)	269	531	1115	2081	3881
FFT rate (Hz)	2130	2137	2162	2080	2200
total rate (Hz)	116	197	324	435	567
avg. masked frames (%)	0	93	93	93	93

Table 2: Breakdown performance figures for our test block FIR reverberation pipeline. All values are time-averaged over a processing-time equal to the duration of the input data.

As can be seen from the results, our approach can bring FIR-based reverberation to the efficiency level of IIR-based techniques [19] with an equivalent cost of about a few tens of operations/time sample. Our masking strategy also appears to be more efficient

than the simpler hearing-threshold cut-off presented in [11] with a reverberation cut-down of up to 93%. Although the degradation becomes noticeable at low processing rates, we found the overall perceived room-effect to be well preserved. We believe an optimized implementation of our approach, including specific assembly language processing and FFT code (we used a simple implementation from [20]) could outperform efficient convolution engines such as [21] for applications where lossy processing is acceptable.

6.3. Massive spatial audio rendering

We finally applied our technique to spatial audio rendering. In this case, we perform a complex multiply on our data to account for various filtering effects (e.g. Head Related Transfer Function [22], atmospheric scattering, occlusions, source directivity) and to delay the audio data according to the distance and direction of each incoming source signal. As in the reverberation application, our pipeline computes a binaural output and thus requires four inverse FFTs to reconstruct a full output frame of stereo audio data. Table 3 shows results of our approach when used to process 1000 3D sound sources, using a variable number of coefficients for the entire pipeline. In our test case, all sound sources are instances of 8 primary audio signals, demonstrating possible application to auralizing sound reflection or diffraction resulting from a geometrical acoustics simulation [23, 24]. Refresh rates for a simultaneous basic 3D-graphics rendering of the sound sources are also provided. As we reduce the number of audio operations, more CPU-time can be devoted to graphics rendering resulting in increased frame-rates. For this application, our approach can lead to 8 to 15 fold-improvement over the brute force techniques while maintaining good output quality. In particular, we estimate that our unoptimized approach can render up to 6 times as many 3D sound sources as the approach of [8] which relied on SSE-optimized code and hardware spatialization. Besides, it does not require specific spatial clustering. Example movie files demonstrating these results are also available on-line.

target FFT bins	all (512000)	80000	50000	10000
audio rate (Hz)	8	56	70	150
graphics rate (Hz)	n.a.	8	15	27
avg. masked frames (%)	0	24	24	24

Table 3: Performance figures for our test 3D audio-visual rendering application processing 1000 sound sources. Masking is turned-on for all test-cases except the reference (i.e., first column). Results are averaged over 11 sec. of processing-time.

7. DISCUSSION

One limitation of our importance sampling scheme is that it requires a fine-grain scalable model of sounds to be applicable. However, we also experimented with coarser-grain time-domain representations with promising results [14]. As with all frequency-domain processing approaches, it might require many inverse FFTs per frame to reconstruct multiple channels of output. This might be a limiting factor for applications requiring multi-channel output (e.g., 5.1 surround). However, in most cases the number of output channels is small.

Another limiting factor is that we use pre-computed information to limit the overhead of our frequency domain processing and final reconstruction. In the case were the input signals are not known in advance (e.g., voice over IP, real-time synthesis,...), an equivalent representation would have to be constructed on-the-fly prior to processing. We believe that if a small number of such streams are present, our approach would still improve the overall performance.

Pre-sorting the FFT data also implies that the processing should not drastically affect the frequency spectrum of the input signals which might not be the case. However, any approach using signals encoded with a lossy algorithm would suffer from the same limitation since perceptual (e.g., masking) effects would be encoded *a priori*. A solution to the problem would be to store sorted FFT data associated to a number of subbands and re-order them in real-time according to how filtering operations might affect the level in each subband. Sorting the output frequency-domain data would also be necessary if several effects have to be chained together. Another solution to better account for filtering effects would be to extend our importance sampling strategy to account explicitly for frequency content (currently, importance is implied by the pre-computed ordering of STFT data so that only the number of processed bins has to be determined). This might yield to an approach closer in spirit to [5] albeit we would still benefit from scalability and masking estimation (and would not require specific filter representation).

Although it can be optimized, our STFT data is not a compact representation (at least not as compact as standard perceptually coded representations, such as mp3 or AAC) which could be a limitation for streaming or bus transfers. However, masking calculations limit this problem, especially since they only require the additional descriptors to evaluate audible data.

Finally, the importance model used for selecting which FFT coefficients to process could be improved by using a more sophisticated loudness-related metric.

8. CONCLUSIONS

We presented a scalable approach to efficiently filter and mix-down a large number of audio signals in real-time. By pre-computing a Fourier frequency-domain representation of our input audio data augmented with a set of audio descriptors, we are able to concentrate our processing efforts on the most important components of the signals. In particular, we show that we can identify signals which will not be audible in the output at each processing time-frame. Such signals can be discarded thus reducing computational load and bus traffic. Remaining audible signals are sampled based on an importance metric so that only a subset of their representation is processed to produce a frame of output. Our approach yields a 3 to 15-fold improvement in overall processing rate compared to brute-force techniques with minimal degradation of the output. As future extensions, we plan to conduct perceptual validation studies to assess the auditory transparency of our approach at several "processing bit-rates" and further improve on our masking calculation and importance sampling metrics.

9. ACKNOWLEDGMENTS

The author would like to thank the anonymous reviewers for their useful comments. This research was supported in part by the 2-year RNTL project OPERA, co-funded by the French Ministry of Research and Ministry of Industry.
http://www-sop.inria.fr/reves/OPERA.

10. REFERENCES

[1] E. M. Painter and A. S. Spanias, "Perceptual coding of digital audio," *Proceedings of the IEEE*, vol. 88, no. 4, Apr. 2000.

[2] Jürgen Herre, "Audio coding - an all-round entertainment technology," in *Audio Engineering Society 22nd International Conference on Virtual, Synthetic and Entertainment Audio (AES'22), Espoo, Finland*, June 15-17 2002, pp. 139–148.

[3] C. A. Lanciani and R. W. Schafer, "Psychoacoustically-based processing of MPEG-I layer 1-2 encoded signals," in *Proc. IEEE Signal Processing Society 1997 Workshop on Multimedia Signal Processing*, June 1997, pp. 53–58.

[4] Chris A. Lanciani and Ronald W. Schafer, "Subband-domain filtering of MPEG audio signals," in *Proceedings of Intl. Conf. on Acoustics, Speech and Signal Processing*, Mar. 1999, pp. 917–920.

[5] Abdellatif B. Touimi, "A generic framework for filtering in subband domain," in *In Proceeding of IEEE 9th Workshop on Digital Signal Processing, Hunt, Texas, USA*, October 2000.

[6] Abdellatif B. Touimi, Marc Emerit, and Jean-Marie Pernaux, "Efficient method for multiple compressed audio streams spatialization," in *In Proceeding of ACM 3rd Intl. Conf. on Mobile and Ubiquitous multimedia*, 2004.

[7] D. Darlington, L. Daudet, and M. Sandler, "Digital audio effects in the wavelet domain," in *Proceedings of COST-G6 Conference on Digital Audio Effects, DAFX2002, Hamburg, Germany*, Sept. 2002.

[8] Nicolas Tsingos, Emmanuel Gallo, and George Drettakis, "Perceptual audio rendering of complex virtual environments," *ACM Transactions on Graphics (Proceedings of SIGGRAPH'04)*, vol. 23, no. 3, 2004.

[9] Mathieu Lagrange and Sylvain Marchand, "Real-time additive synthesis of sound by taking advantage of psychoacoustics," in *Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-01), Limerick, Ireland, December 6-8*, 2001.

[10] Kees van den Doel, Dave Knott, and Dinesh K. Pai, "Interactive simulation of complex audio-visual scenes," *Presence: Teleoperators and Virtual Environments*, vol. 13, no. 1, 2004.

[11] Wen-Chieh Lee, Chung-Han Yang, Chi-Min Liu, and Juin-In Guo, "Perceptual convolution for reverberation," in *In Proceeding of 115th AES Convention, Los Angeles*, Oct. 2003.

[12] Udo Zölzer, Ed., *DAFX - Digital Audio Effects*, Wiley, 2002.

[13] Perfecto Herrera, Xavier Serra, and Geoffroy Peeters, "Audio descriptors and descriptors schemes in the context of MPEG-7," in *Procedings of International Computer Music Conference (ICMC99)*, 1999.

[14] Emmanuel Gallo, Guillaume Lemaitre, and Nicolas Tsingos, "Prioritizing signals for selective real-time audio processing," in *proceedings of Intl. Conf. on Auditory Display (ICAD) 2005, Limerick, Ireland*, July 2005.

[15] E. Zwicker and H. Fastl, *Psychoacoustics: Facts and Models*, Springer, 1999, Second Upadated Edition.

[16] Barry D. Kulp, "Digital equalization using fourier transform techniques," in *In Proceeding of 85th AES Convention, Los Angeles*, Nov. 1988.

[17] Jia-Sen Soo and Khee K. Pang, "Multidelay block frequency domain adaptive filter," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 38, no. 2, Feb. 1990.

[18] G. García, "Optimal filter partition for efficient convolution with short input/output delay," in *In Proceeding of 113th AES Convention, Los Angeles*, Oct. 2002.

[19] Mark Kahrs and Karlheinz Brandenburg, Eds., *Applications of Digital Signal Processing to Audio and Acoustics*, Kluwer Academic Publishers, 1998.

[20] William Press, Saul Teukolsky, William Vetterling, and Brian Flannery, *Numerical Recipes in C, 2nd edition*, Cambridge University Press, New York, USA, 1992.

[21] Anders Torger, "BruteFIR software convolution engine," http://www.ludd.luth.se/ torger/brutefir.html.

[22] Durand R. Begault, *3D Sound for Virtual Reality and Multimedia*, Academic Press Professional, 1994.

[23] J.B. Allen and D.A. Berkley, "Image method for efficiently simulating small room acoustics," *J. of the Acoustical Society of America*, vol. 65, no. 4, 1979.

[24] N. Tsingos, T. Funkhouser, A. Ngan, and I. Carlbom, "Modeling acoustics in virtual environments using the uniform theory of diffraction," *ACM Computer Graphics, SIGGRAPH'01 Proceedings*, pp. 545–552, Aug. 2001.

Progressive Perceptual Audio Rendering of Complex Scenes

Thomas Moeck[1,2] Nicolas Bonneel[1] Nicolas Tsingos[1] George Drettakis[1] Isabelle Viaud-Delmon David Alloza
[1]REVES/INRIA Sophia-Antipolis CNRS-UPMC UMR 7593 EdenGames
[2]Computer Graphics Group, University of Erlangen-Nuremberg

Figure 1: Left: A scene with 1815 mobile sound sources. Audio is rendered in realtime with our progressive lossy processing technique using 15% of the frequency coefficients and with an average of 12 clusters for 3D audio processing. Degradations compared to the reference solution are minimal. Right: a scene with 1004 mobile sound sources, running with 25% of the frequency coefficients and 12 clusters.

Abstract

Despite recent advances, including sound source clustering and perceptual auditory masking, high quality rendering of complex virtual scenes with thousands of sound sources remains a challenge. Two major bottlenecks appear as the scene complexity increases: the cost of clustering itself, and the cost of pre-mixing source signals within each cluster.

In this paper, we first propose an improved hierarchical clustering algorithm that remains efficient for large numbers of sources and clusters while providing progressive refinement capabilities. We then present a lossy pre-mixing method based on a progressive representation of the input audio signals and the perceptual importance of each sound source. Our quality evaluation user tests indicate that the recently introduced audio saliency map is inappropriate for this task. Consequently we propose a "pinnacle", loudness-based metric, which gives the best results for a variety of target computing budgets. We also performed a perceptual pilot study which indicates that in audio-visual environments, it is better to allocate more clusters to visible sound sources. We propose a new clustering metric using this result. As a result of these three solutions, our system can provide high quality rendering of thousands of 3D-sound sources on a "gamer-style" PC.

Keywords: Audio rendering, auditory masking, ventriloquism, clustering

1 Introduction

Spatialized audio rendering is a very important factor for the realism of interactive virtual environments, such as those used in computer games, virtual reality, or driving/flight simulators, etc. The complexity and realism of the scenes used in these applications has increased dramatically over the last few years. The number of objects which produce noise or sounds can become very large, e.g., cars in a street scene, crowds in the stadium of a sports game etc. In addition, recent sophisticated physics engines can be used to synthesize complex sound effects driven by the physics, for example the individual impact sounds of thousands of pieces of a fractured object.

Realistic 3D audio for such complex sounds scenes is beyond the capabilities of even the most recent 3D audio rendering algorithms. The computational bottlenecks are numerous, but can be grouped into two broad types: the cost of *spatialization*, which is related to the audio restitution format used; and the *per sound source* cost, which relates to the different kinds of effects desired. An example of the former bottleneck is Head Related Transfer Function (HRTF) processing for binaural rendering [Møller 1992], or the rendering of numerous output channels for a Wave Field Synthesis (WFS) system [Berkhout et al. 1993], which can use hundreds of speakers. The latter bottleneck includes typical spatial effects such as delays, the Doppler effect, reverberation calculations, but also any kind of studio effect the sound engineers/designers wish to use in the application at hand.

Recent research has proposed solutions to these computational limitations. Perceptual masking with sound source clustering [Tsingos et al. 2004], or other clustering methods [Herder 1999; Wand and Straßer 2004] do resolve some of the issues. However, the clustering algorithms proposed to date are either restricted to static scenes, or add an unacceptable computation overhead due to a quadratic step in cluster construction when the number of sources is large. In addition, the cost of per source computation, sometimes called premixing, can quickly become a bottleneck, again for complex soundscapes.

In this paper we present two contributions:

- The first contribution is the speed improvement of the clustering and premixing steps of the [Tsingos et al. 2004] approach. First, we introduce a new recursive clustering method, which can operate with a fixed or variable target cluster budget, and thus addresses the issue of spatialization cost mentioned above. Second, we develop a novel "pinnacle-based" scalable premixing algorithm, based on [Tsingos 2005], pro-

viding a flexible, perceptually-based framework for the treatment of the per-source computation costs.

- The second contribution is the investigation of perceptual issues related to *clustering* and *premixing*, based on *pilot user studies* we conducted. In particular, we investigate the influence of visuals on audio clustering for audio-visual scenes, and propose a modified metric taking into account the indication that it is probably better to have more sources in the view frustum. For scalable premixing, we evaluated the different metrics which can be used, including recently developed perceptual models, such as the audio saliency map [Kayser et al. 2005], and performed a perceptual quality test for the new algorithm.

In the following, we briefly overview related previous work, including that in the acoustics and perception literature. We then present the new recursive clustering method in Sect. 3, the new study and the resulting algorithm for scalable premixing in Sect. 4 and the study and novel metric for crossmodal audio-visual clustering in Sect. 5. We present some implementation issues and results, then conclude with a discussion of our approach.

2 Previous work

Relatively little effort has been devoted to the design of scalable rendering strategies for 3D audio, which can provide level-of-detail selection and graceful degradation. We give below a short overview of the previous work most relevant to our problem.

Encoding and rendering of spatial auditory cues. Progressive spatial sound encoding techniques can be roughly subdivided in two categories.

A first category is based on a physical reconstruction of the wavefield at the ears of the listener. For instance, several approaches have been proposed to perform progressive binaural rendering [Blauert 1997] by decomposing HRTFs on a set of basis functions through principal component analysis [Chen et al. 1995; Larcher et al. 2000; Jot and Walsh 2006]; while providing level-of-detail, they are limited to this unique restitution format. Alternatively, decomposition of the wavefield can be used (e.g., spherical harmonics) [Malham and Myatt 1995] for level-of-detail, relatively independently of restitution setup (e.g., conversion to binaural format [Jot et al. 1999]). High accuracy requires a large number of channels however, limiting the methods' applicability.

Another category performs world-space compression of positional cues by clustering nearby sound sources and using a unique representative position per cluster for spatial audio processing [Herder 1999; Wand and Straßer 2004; Tsingos et al. 2004]. Wand et al. [Wand and Straßer 2004] group sources in a hierarchical spatial data structure as used for point-based rendering. However, their approach is very efficient only in the case of static sources. Tsingos et al. [Tsingos et al. 2004] recently introduced a clustering algorithm driven by a loudness-weighted geometrical cost-function (see Sec. 3). They also use precomputed descriptors (e.g., loudness and tonality) to sort the sources by decreasing importance and perform a greedy culling operation by evaluating auditory masking at each time-frame of the simulation. Inaudible sources can then be safely discarded. Although this approach was found to perform well for environments containing a few hundred sources, it is unclear that it scales well to larger numbers of sources and clusters due to the cost of the proposed clustering algorithm which, in their case, implies a near quadratic number of evaluations of the cost-function.

Scalable audio processing. Fouad et al. [Fouad et al. 1997] propose a level-of-detail progressive audio rendering approach in the time-domain; by processing every n-*th* sample, artifacts are introduced at low budget levels. Wand and Straßer [Wand and Straßer 2004] introduce an importance sampling strategy using random selection, but ignore the signal properties, thus potentially limiting the applicability of this method.

A family of approaches has been proposed to directly process perceptually-coded audio signals [Lanciani and Schafer 1997; Lanciani and Schafer 1999; Darlington et al. 2002; Touimi 2000; Touimi et al. 2004] yielding faster implementations than a full decode-process-recode cycle. Although they are well suited to distributed applications involving streaming over low-bandwith channels, they require specific coding of the filters and processing. Moreover, they cannot guarantee efficient processing for a mixture of several signals, nor that they would produce an optimal result. Other methods have explored how to extend these approaches by concurrently prioritizing subparts of the original signals to process to guarantee a minimal degradation in the final result [Gallo et al. 2005; Tsingos 2005; Kelly and Tew 2002]. Most of them exploit masking and continuity illusion phenomena [Kelly and Tew 2002] to remove entire frames of the original audio data in the time-domain [Gallo et al. 2005] or, on a finer scale, process only a limited number of frequency-domain Fourier coefficients [Tsingos 2005].

Crossmodal studies. While the primary application of 3D audio rendering techniques is simulation and gaming, no spatial audio rendering work to date evaluates the influence of combined visual and audio restitution on the required quality of the simulation. However, a vast amount of literature in neurosciences suggest that cross-modal effects, such as ventriloquism, might significantly affect 3D audio perception [Hairston et al. 2003; Alais and Burr 2004]. This effect tells us that in presence of visual cues, the location of a sound source is perceived shifted toward the visual cue, up to a certain threshold of spatial congruency. Above this threshold, there is a conflict between the perceived sound location and its visual representation and the ventriloquism effect no longer occurs. The spatial window (or angular threshold) of this effect seems to depend on several factors (e.g., temporal synchronicity between the two channels and perceptual unity of the bimodal event) and can vary from a few degrees [Lewald et al. 2001] up to 15° [Hairston et al. 2003].

3 Optimized Recursive Clustering

In previous work [Tsingos et al. 2004], large numbers of sound sources are dynamically grouped together in clusters, and a single new *representative* point source is created. While this approach allows the treatment of several hundred sound sources on a standard platform, it does incur some computational overhead, which becomes a potential bottleneck as a function of the number of input-sources/target-clusters and the available computational budget. To resolve this limitation, we next present a new recursive clustering algorithm. For improved efficiency and accuracy we propose two recursive algorithms: One with a fixed budget of clusters and one with a variable number of clusters. The fixed budget approach allows us to set a fixed computational cost for 3D audio processing and is also useful to address a fixed number of 3D-audio hardware channels when available for rendering. The variable number of clusters dynamically adjusts the computational cost to obtain the best trade-off between quality and speed in each frame, given a user-specified error threshold.

We also discuss a variant of this algorithm which has been included in the commercially available game "Test Drive Unlimited" by EdenGames/ATARI.

In the method of [Tsingos et al. 2004], sources are grouped together

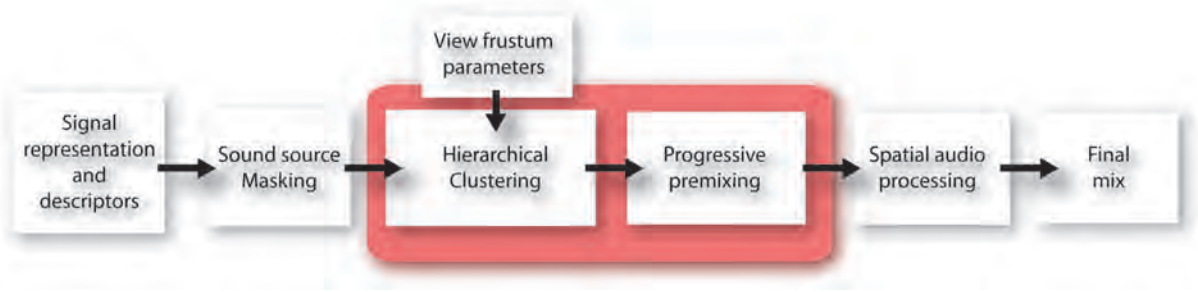


Figure 2: Overview of our overall sound rendering pipeline. In particular, we introduce an improved hierarchical sound source clustering that better handles visible sources and a premixing technique for progressive per-source processing.

by using a clustering algorithm based on the Hochbaum-Shmoys heuristic [Hochbaum and Schmoys 1985]. First, this strategy selects n cluster representatives amongst the k original sources by doing a farthest-first traversal of the point set. The cost-function used is a combination of angular and distance errors to the listener of the candidate representative source with respect to the sources being clustered. An additional weighting term, based on each source's instantaneous loudness value, is used to limit error for loud sources. For details see [Tsingos et al. 2004].

3.1 Recursive fixed-budget approach

In a first pass, we run the original clustering algorithm with a target number of clusters n_0. In each subsequent pass, every generated cluster gets divided into n_k clusters. The total budget of clustering is thus $\prod_k n_k$. The original clustering algorithm can be considered as a special case where only n_0 is set. In our tests, we typically used a two-level recursion.

3.2 Variable cluster budget

This approach dynamically allocates the number of clusters in realtime. This is especially useful for scenes where sounds are frequently changing during time in shape, energy as well as in location. The algorithm then flexibly allocates the required number of clusters; thus clusters are not wasted where they are not needed.

First, every sound source which has not been masked [Tsingos et al. 2004], is put in one cluster which is then recursively split into two until an appropriate condition is met. In every recursion step the error in angle relative to the listener position is computed, for each sound source in a cluster, relative to its centroid. If the average angle error is below a threshold, cluster splitting is terminated. In our tests, we found that a 25° threshold value proved satisfactory.

3.3 Quantitative error/performance comparison

We have performed a quantitative comparison between the previous clustering approach of [Tsingos et al. 2004], and the two proposed recursive clustering methods. We ran several thousand tests with random configurations of 800 sound sources, using the different algorithms and measuring the time and the error (in terms of distance and angle) for each method.

Figure 3 shows the results of the tests for fixed budgets of 12 and 6 clusters using different two-level subdivision strategies. For instance, line 3/4 corresponds to a clustering using 12 clusters (3 top-level clusters recursively refined into 4). The line 12/1 corresponds

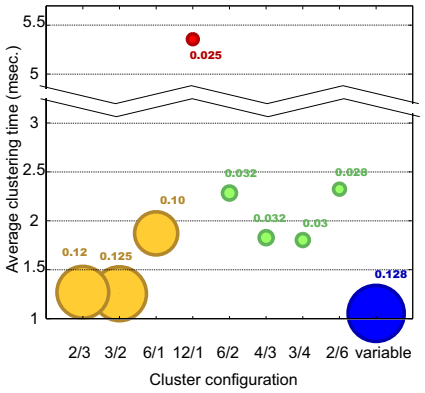


Figure 3: Benchmarks for hierarchical clustering of 800 sources using different 6 and 12 cluster configurations. We display average clustering error (also denoted by circle size). Note the significant speed-up compared to the non-hierarchical algorithm for the 12 cluster configurations (in red) while the errors remain similar.

to the previous clustering approach of [Tsingos et al. 2004] where all 12 clusters are top level.

As we can see, the performance of the recursive approaches are clearly better than the direct algorithm. For the same final budget, the 3/4 and 4/3 configurations appear to be better choices in terms of speed. As expected the error is larger for hierarchical clustering since it leads to less optimal cluster placement. However, as the number of clusters grows this effect tends to disappear. The variable cluster method is faster on average. However, with our current settings it also created fewer clusters (6.6 clusters created on average) and, as a consequence, has higher average error. Interestingly, the peak number of clusters created by the variable method is 22, which underlines the flexibility and adaptability of the approach.

To limit the error compared to a non-hierarchical implementation, it is preferable to create more top level clusters. For instance a 4/3 split is better than a 2/6 split for a 12 cluster configuration, altough it might be slightly slower. Hence, this choice depends on the desired time vs. error tradeoff for the application at hand.

3.4 Implementation in a commercial game

The audio clustering technique was used in the development of the commercially available computer game *Test Drive Unlimited*. In this car racing game, the sound of each racing vehicule is synthesized based on numerous mechanical quantities. The sound emitted by each wheel is controlled by 20 physical variables while 8 variables control the engine/transmission sounds. Four additional vari-

Figure 4: Sound source clustering in the *Test Drive Unlimited* engine. Red wireframe spheres are clusters. ©Eden Games-ATARI 2006.

ables control aerodynamic phenomena. These variables are used for real-time control and playback of a set of pre-recorded sound samples. All sound sources are then rendered in 5.1 or 7.1 surround sound. For implementation on the *XBOX360*, Eden Games adopted a variant of the recursive variable budget technique described above. In particular, a budget of 8 clusters was used at each recursion level. If the quality criterion is not met for this budget, a local clustering step is applied in each cluster. However, the local nature of clustering resulted in audible artifacts from one frame to the next, because of sources moving from one cluster to another. To resolve this problem, sources are ordered by perceptual priority, and the most important ones will prefer to be clustered with the cluster of the previous frame, effecting a form of temporal coherence.

Despite this improvement, extreme cases still presented some difficulty, notably the case of a car crashing into an obstacle. In this case, the physics engine generates numerous short-lasting sound sources in the immediate neighbourhood of the vehicle. Temporal coherence is thus useless in this context. The solution to this issue is to apply a separate clustering step to the sources generated by physics events; this results in more clusters overall, but resolves the problems of quality.

A snapshot of *Test Drive Unlimited* with a visualisation of the sound source clusters superimposed in red, is shown in Figure 4.

4 Scalable perceptual premixing

In order to apply the final spatial audio processing to each cluster (see Figure 2), the signals corresponding to each source must first be *premixed*. The premixing stage can be as simple as summing-up the signals of the different sources in each cluster. In addition, a number of audio effects usually have to be applied on a per-source basis. Such effects include filtering (e.g., distance, occlusions), pitch-shifting (e.g., Doppler effect) or other studio-like effects [Zölzer 2002]. Hence, when large numbers of sound sources are still present in the scene after auditory culling (masking) or for systems with limited processing power, the cost of this stage can quickly become the bottleneck of the audio rendering pipeline.

In this section, we propose a progressive signal processing technique that can be used to implement scalable per-source processing. Our work is based on the approach of [Tsingos 2005] which we briefly summarize in Figure 5.

This approach uses a specific time-frequency representation of audio signals. At each time-frame (typically 1024 samples at 44.1KHz), the complex-valued coefficients of a short-time Fourier transform (STFT) are precomputed and stored in decreasing modulus order. In real-time during the simulation, the algorithm prioritizes the signals and allocates to each source a number of coefficients to process, so that a predefined budget of operations is

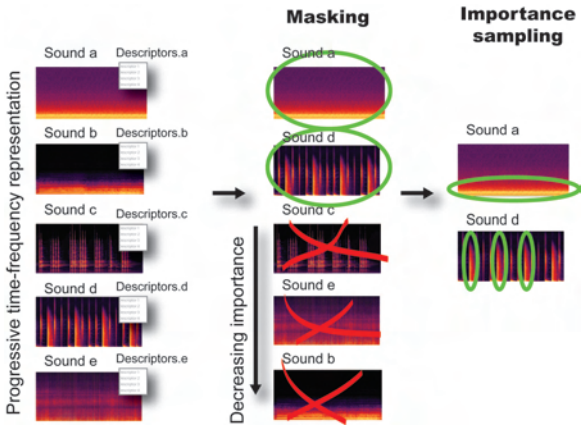


Figure 5: Overview of our progressive perceptual premixing.

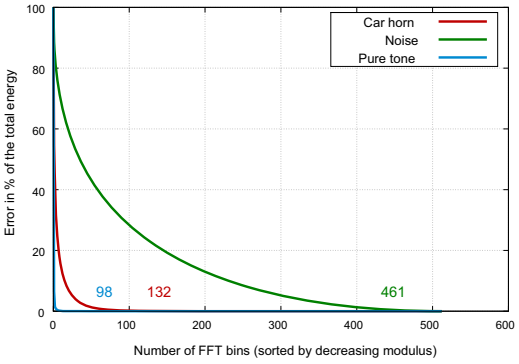


Figure 6: Reconstruction error as a function of target FFT bins. Tonal signals require fewer coefficient than noisier signals since their Fourier representation is much sparser. The value next to each curve corresponds to our pinnacle value. In [Tsingos 2005], the integral of these curves is used to measure the coding efficiency for each frame of input signal.

respected. In [Tsingos 2005], this importance sampling stage is driven by the energy of each source at each time-frame and used to determine the cut-off point in the list of STFT coefficients. However, using only energy for importance sampling leads to sub-optimal results since it does not account for the sparseness of the representation obtained for each signal. For instance, a loud tonal signal might require fewer coefficients than a weaker noisier signal for transparent reconstruction (Figure 6). An additional weighting term measuring the efficiency of coding for each frame of input signal was thus proposed for budget allocation.

In the following, we introduce an improved budget allocation strategy. We also present the results of a perceptual quality study aimed at evaluating our novel technique and several possible importance metrics used for prioritizing source signals.

4.1 Improved budget allocation

The pinnacle value. Contrary to [Tsingos 2005], our improved budget allocation pre-computes the explicit number of STFT coefficients necessary to transparently reconstruct the original signal. This value, that we call *pinnacle*, is pre-computed for each time-frame of input audio data and stored together with the progressive

STFT representation. To compute the pinnacle value, we first sort the STFT coefficients by decreasing modulus order. The energy of each coefficient is integrated until a threshold of at least 99.5% of the total energy of the frame is reached and the number of corresponding coefficients is greater than $(1-tonality)N/2$, where N is the total number of complex Fourier coefficients in the frame and $tonality \in [0,1]$ is the tonality index of the frame [Painter and Spanias 2000; Kurniawati et al. 2002]. This index is close to 1 for tonal signals and drops to 0 for noisier signals.

Iterative importance sampling. We assume a constant number of arithmetic operations will be required for each complex STFT coefficient. Hence, fitting a budget number of operations for our pipeline at each processing frame directly amounts to selecting a budget number of coefficients for each frame of input sound signal. We can take advantage of pre-storing our FFT in decreasing energy order by directly processing the n_i first coefficients for each input signal s_i, so that the sum of all n_is does not exceed our total budget N. To determine the n_is, we first assign an importance value to each signal. This importance value can typically be the energy or loudness of the signal as proposed in [Tsingos et al. 2004; Tsingos 2005]. In this work, we also experimented with a *saliency* value derived from the model recently proposed in [Kayser et al. 2005]. This model is very similar to the visual saliency maps [Itti et al. 1998] but it is applied on a time-frequency domain representation of audio signals. In our case, after computing the auditory saliency map, we integrated saliency values over a small number of frequency subbands (we typically use 4 on a non-linear frequency scale).

Then, every input signal gets assigned a number of bins n_i relative to its relative importance as follows:

$$n_i = I_i / \sum_i I_i \cdot \text{targetCoeffs} \quad (1)$$

where I_i is the importance of the source i. Ideally, n_i should be smaller than the signal's pinnacle value to avoid suboptimal budget allocation as can be the case with the approach of [Tsingos 2005]. To avoid such situations, all remaining coefficients above pinnacle threshold are re-assigned to the remaining signals, that do not already satisfy the pinnacle value criterion. This allocation is again relative to the importance values of the signals:

$$n_i += I_i / \sum_i I_i \cdot \text{extraCoeffs} \quad (2)$$

The relative importance of each remaining signal is updated according to the reallocation of coefficients. If the budget is high enough, the process is iterated until all signals satisfy the pinnacle criteria or receive a maximal number of coefficients.

4.2 Quality evaluation study

To evaluate possible importance metrics and evaluate our pinnacle-based algorithm we conducted a quality evaluation study.

Experimental procedure. Seven subjects aged from 23 to 40 and reporting normal hearing volunteered for five different test sessions. For each session, we used a *Multiple Stimuli with Hidden Reference and Anchors* procedure (MUSHRA, ITU-R BS.1534) [Stoll and Kozamernik 2000; EBU 2003; International Telecom. Union 2001-2003]. Subjects were asked to simultaneously rank a total of 15 stimuli relative to a reference stimulus on a continuous 0 to 100 quality scale. The highest score corresponds to a signal indistinguishable from the reference. The reference stimuli were different mixtures of ambient, music and speech signals. In all cases, 12 of the 15 test-stimuli consisted of degraded versions of the mixture computed using our progressive mixing algorithm at various budgets (5%, 10% and 25% for music and ambient and 2%, 5% and 10% for speech), using our pinnacle-based technique, not using the pinnacle and using either loudness or saliency-based prioritization. In all cases, our processing is done using 32-bit floating point arithmetic and reconstructs signals at 44.1KHz. Two anchor stimuli, providing reference degradations, were also included. In our case, we chose a downsampled 16KHz/16-bit version of the mixture and a *mp3*-encoded version at 64Kbps. Finally, a hidden reference was also included. Stimuli were presented over headphones. Subjects could switch between stimuli at any point while listening. They could also define looping regions to concentrate on specific parts of the stimuli. A volume adjustment slider was provided so that subjects could select a comfortable listening level.

Results. Our study confirmed that the scalable processing approach is capable of generating high quality results using 25% of the original audio data and produces acceptable results with budgets as low as 10%. In the case of speech signals, for which the STFT representation is sparser, the algorithm could generate an acceptable mixture (avg. score 56/100) with only 2% of the original coefficients. As can be seen on Figure 7 (left), our approach yields significantly better results than a 16KHz reference signal (16KHz processing would correspond to a 30% reduction of data compared to our 44.1KHz processing). At 25% budget (or 10% in the case of speech), we obtain results comparable or better than the 64Kbps mp3-encoded reference. We performed an analysis of variance (ANOVA) [Howell 1992] on the results. As expected, the analysis confirmed a significant effect of the computing budget ($p<0.01$) on the quality of the resulting signal (Figure 7 right). We can see that the variation of perceived qualty is not generally a linear function of budget, especially for more tonal signals (music, speech) which can be efficiently encoded until a sharp breakdown point is reached. Interaction between budget and importance metric was found to be significant ($0.05<p<0.01$). At low or high budgets, the two metrics lead to very similar results. However, for intermediate budgets, loudness-based prioritization improved perceived quality relative to the saliency-based alternative. Similarly, using our new pinnacle algorithm also leads to a slight improvement in the results, especially for cases where both tonal and noisier signals are present. Noisier stationary sounds, which do not contain strong spectral features, usually receive a lower saliency value although they might contain significant energy and require more coefficients to be properly reconstructed. We believe this might explain why saliency-based prioritization led to lower perceived quality in some cases.

5 Cross-modal effects for sound scene simplification

In the preceding sections, we have improved different aspects of audio rendering for complex scenes, without consideration for the corresponding visuals. Intuitively, it would seem that such interaction of visual and audio rendering should be taken into account, and play a role in the choice of metrics used in the audio clustering algorithm. A first attempt was presented in [Tsingos et al. 2004], but was inconclusive presumably due to the difficulties with speech stimuli, which are generally considered to be a special case.

Research in ventriloquism (see Section 2), could imply that we should be more tolerant to localization errors for sound rendering when we have accompanying visuals. If this were the case, we could change the weighting terms in the clustering algorithm to create fewer clusters for sound sources in the visible frustum. However, a counter argument would be that in the presence of visuals,

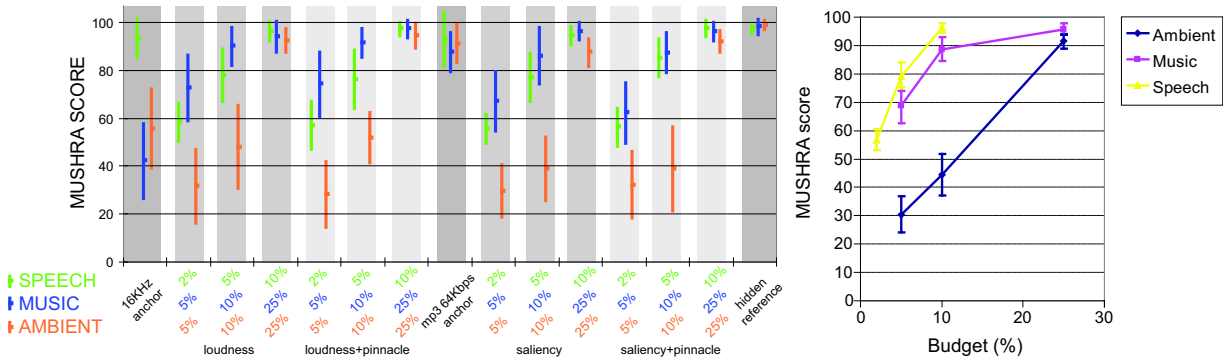


Figure 7: Left: Average MUSHRA scores and 95% confidence intervals for our progressive processing tests. Right: Average MUSHRA scores and 95% confidence intervals as a function of budget. Note how perceived quality does not vary linearly with the processing budget and also varies depending on the type (i.e., sparseness) of the sounds.

we are more sensitive to localization, and we should favour more clusters in the viewing frustum.

Our goal was to see whether we could provide some insight into this question with a pilot perceptual study. The next step was to develop and test an improved audio clustering algorithm based on the indications obtained experimentally.

5.1 Experimental setup and methodology

We chose the following experimental setup to provide some insight on whether we need more clusters in the visible frustum or not.

The subjects are presented with a scene composed of 10 animated - but not moving - objects emitting "ecologically valid" sounds, i.e., a moo-ing sound for the cow, a helicopter sound, etc. (Figure 8; also see and hear accompanying video).

We have two main conditions: audio only (i.e., no visuals) (condition *A*) and audio-visual (*AV*). Within each main condition we have a control condition, in which sources follow a uniform angular distribution, and the condition we test, where the proportion of clusters in the visible frustum and outside the visible frustum is varied.

We ran our test with 6 subjects (male, aged 23-45, with normal or corrected to normal vision, reporting normal hearing). All were naive about the experiment. Five of them had no experience in audio. Prior to the test, subjects were familiarized with isolated sound effects and their corresponding visual representation.

The subject stands 1 meter away from a 136 x 102 cm screen (Barco Baron Workbench), with an optical headtracking device (ART) and active stereo glasses (see the video). The field of view in this large screen experiment is approximately 70 °.

Headphones are used for audio output and our system uses binaural rendering [Blauert 1997; Møller 1992] using the LISTEN HRTF database (http://recherche.ircam.fr/equipes/salles/listen/). Our subjects were not part of the database. Hence, they performed a "point and click" pre-test to select the best HRTFs over a subset of 6 HRTF selected to be "most representative" similar to [Sarlat et al. 2006]. The marks attributed for the test are given with a joystick.

The *A* condition was presented first for three candidates, while *AV* condition was presented first for the other three. No significant effect of ordering was observed.

To achieve the desired effect, objects are placed in a circle around the observer; 5 are placed in the viewing frustum and 5 outside. For both control and main conditions, four configurations are used

Figure 8: An example view of the experimental setup for the audio-visual pilot user study.

randomly, by varying the proportion of clusters. Condition 1/4 has one cluster in the view frustum and 4 outside, 2/3, has 2 in the view frustum and 3 outside, etc. A uniform distribution of clusters corresponds to condition 1/4, with only 1 cluster in the frustum. Each condition is repeated 15 times with randomized object positions; these repetitions are randomized to avoid ordering effects.

We used the ITU-recommended *triple stimulus, double blind with hidden reference* technique [and 1993; ITU-R 1994]: 2 versions of the scene were presented ("A" and "B") and a given reference scene which corresponds to unclustered sound rendering. One of the 2 scenes was always the same as the reference (a *hidden reference*) and the other one corresponds to one of our clustering configurations. For each condition, the subject was presented with a screen with three rectangles ("A", "R" and "B"), shown in Fig. 8. The subjects were given a gamepad, and were instructed to switch between "A", "B" and "R" using three buttons on the pad, which were highlighted depending on the version being rendered. The subjects were asked to compare the quality of the approximations ("A" or "B") compared to the reference. They were asked to perform a "*quality judgment paying particular attention to the localization of sounds*" for the 2 test scenes, and instructed to attribute one of 4 levels of evaluation "No difference", "Slightly different", "Different" and "Clearly different" from the reference, which were indicated in rectangles next to the letter indicating the scene version (see Fig. 8 and accompanying video).

5.2 Analysis and results

We attributed a mark for each evaluation (from 0 to 3). As suggested by this ITU-R standard protocol, we only kept the difference between the test sample and the hidden reference. We also normal-

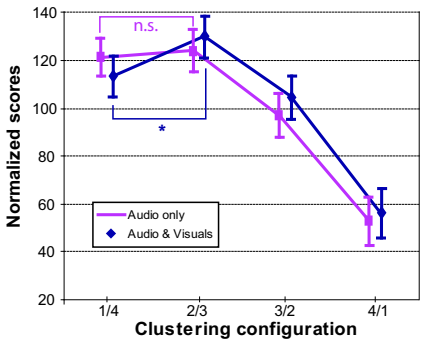


Figure 9: Mean values and 95% confidence intervals (N=6) in *A* and *AV* conditions as a function of the number of clusters inside/outside the view frustrum. For *AV*, the 2/3 configuration gives the best quality scores, which is not the case in the *A* condition. The "*" underlines that quality judgements in 1/4 and 2/3 cluster configurations for *AV* are significantly different ($p<0.05$), while the same comparison is non significant (n.s.) in the *A* condition.

ized the data by dividing each mark by the mean score of the user (the average of all marks of the candidate over all his tests).

There was no significant difference between the *A* and *AV* conditions regarding the respective scores of each cluster configuration. However, the difference of quality ratings between configurations was not similar in the two conditions. In condition *A*, 1/4 and 2/3 configurations lead to a similar quality evaluation (see Figure 9). In condition *AV*, the best quality is perceived in configuration 2/3. While 2/3 and 1/4 configurations are not perceived differently in condition *A* (Wilcoxon test, N=90, T=640.5, Z=0.21, p=0,83), the quality scores of 2/3 configuration are higher than those of 1/4 configuration in condition *AV* (Wilcoxon test, N=90, T=306.5, Z=2.56, p=0.01).

Overall, we consider the above results as a significant indication that, when we use the audio clustering algorithm with visual representation of the sound sources, it is better to have two clusters in the view frustrum, compared to a uniform angular distribution. This is indicated by the results for the 2/3 configuration, which is statistically different from all the other configurations in the AV condition. We expect this effect to be particularly true for scenes where there are visible sound sources in the periphery of the view frustum.

5.3 An audio-visual metric for clustering

Given the above observation, we developed a new weight in the clustering metric which encourages more clusters in the view frustum. We modify the cost-function of the clustering algorithm by adding the following weigthing term:

$$1+\alpha\left(\frac{\cos\theta_s-\cos\theta_f}{1-\cos\theta_f}\right)^n \qquad (3)$$

where θ_s is the angle between the view direction and the direction of the sound source relative to the observer, θ_f is the angular half-width of the view frustum and α controls the amplitude and n decay-rate of this visual improvement factor.

6 Implementation and Results

We ran tests on two scenes, one is a variant of the highway scene from [Tsingos et al. 2004], and another is a city scene. Both scenes

Figure 10: Left: the clusters without the audio-visual metric. Right: the clusters with our new metric. We clearly see that the new metric separates the sources appropriately.

are shown in Figure 1. In both cases, we used a platform with dual-core 3GHz *Xeon* processor and *NVidia 7950GX2* graphics accelerator; our system is built using the Ogre3D graphics engine and our custom audio library. Audio was processed at 44.1KHz using 1024-sample-long time-frames (i.e., 23 msec.). The following tests were performed with masking disabled to get a stable performance measure.

The highway scene contains 1004 sound sources, which are car engine and car stereo music sounds, cow "mooing" sound, train sounds, and water sounds in a stream. We found that a scalable premix budget of 25% is satisfactory in terms of audio quality (please hear and see the accompanying video). Comparing to the reference, we found that our entire perceptual processing pipeline resulted in an average signal-to-interference ratio of 18dB (min=5dB, max=30dB) for the sequence presented in the video. In this scene, clustering took 4.7 msec. per frame. Premixing was very simple and only included distance attenuation and accumulating source signals for each cluster. Premixing using 100% of the original audio data took 6 msec. Using our scalable processing with 25% budget we bring this cost down to 1.83 msec.

The street scene contains 1800 sound sources, which are footstep sounds and voices for the people in the crowd, car engine sounds, car radio sounds, bird sounds and sirens. Again, a scalable premix budget of 15% is satisfactory for this scene. Overall, we measured an average signal-to-interference ratio of 17dB (min=4dB, max=34dB) for the sequence presented in the video. In this scene, clustering took 5.46 msec. per frame while premixing using 100% of the original audio data took 6.84 msec. Using our scalable processing with 15% budget we bring the cost of premixing down to 2.17 msec.

In the commercial game *Test Drive Unlimited*, the average number of simultaneous sound sources is 130. A typical "player car" can generate up to 75 sound sources, while "AI" cars have a simplified model of maximum 42 sources. A maximum of 32 clusters were used in the game, although in a vast majority of cases 10 clusters are sufficient. Overall, the clustering approach discussed in Section 3.4 results in a reduction of 50-60% of CPU usage for the audio engine, which is freed for other application tasks, such as AI, gameplay etc.

To test the new audio-visual criterion, we constructed a variant of the street scene and an appropriate path, in which the positive effect of this criterion is clearly audible. For this test, we used $\alpha = 10$ and $n = 1.5$, which proved to be satisfactory. The scene used can be seen and heard in the video; the user follows a path in the scene (see accompanying video) and stops in a given location in the scene. We have 132 sources in the scene and target budget of 8 clusters. By switching between the reference, and the approximations with and without the audio-visual metric, we can clearly hear the improvement when more clusters are used in the view frustum. In particular, the car on the right has a siren whose sound is audibly displaced towards the centre with the audio-only metric.

7 Discussion and Conclusion

In this paper, we proposed a fast hierarchical clustering approach that can handle large numbers of sources and clusters. We also proposed a progressive processing pipeline for per-source effects (i.e., the premixing) that allows us to choose the best performance/quality ratio depending on the application and hardware constraints. Combined with auditory masking evaluation, these new algorithms allow for real-time rendering of thousands of mobile sound sources while finely controlling processing load vs. quality. In fact, while designing the test examples for the paper, the major problem we faced was authoring environments complex enough and most of the performance limitations actually came from the graphics engine. However, with next-generation games making increased used of procedurally-generated audio (e.g., based on physics engines), scenes with thousands of sound events to process are likely to become common in the near future. In our examples we only used simple per-source processing. However, in most gaming applications each source is likely to be processed with a chain of various effects (e.g., occlusion filters, echoes, re-timing, pitch-shifting, etc.) that would make our scalable approach even more attractive.

We also presented our perceptual studies for clustering and scalable premixing. A cross-modal perceptual study aimed at determining possible influence of the visuals on the required quality for audio clustering. Although one could expect ventriloquism to allow for rendering simplifications for visible sources, our study suggest that more clusters might actually be required in this case. A possible explanation for this is that, in a complex scene, clustering is likely to simplify auditory localization cues beyond common ventriloquism thresholds. As a consequence, we introduced a new metric to augment the importance of sources inside the view frustum. We demonstrated an example where, with a large number of sound sources outside the view frustum, it leads to improved results. We also performed a user-study of quality for the scalable premixing approach and showed that it leads to high quality results with budgets as low as 20 to 15% of the original input audio data. Although saliency-based importance appeared to show some limitations for our scalable processing algorithm compared to loudness, it might still be useful for prioritizing sources for clustering.

In the future, it would be interesting to experiment with auditory saliency metrics to drive clustering and evaluate our algorithms on various combinations of A/V displays (e.g., 5.1 surround or WFS setups). Also, the influence of ventriloquism on these algorithms merits further study. We also believe that authoring is now becoming a fundamental problem. Adapting our algorithms to handle combinations of sample-based and procedurally synthesized sounds seems a promising area of future research.

8 Acknowledgements

This research was funded by the EU IST FET Open project IST-014891-2 CROSSMOD (http://www.crossmod.org). We thank Autodesk for the donation of Maya, P. Richard and A. Olivier-Mangon for modelling/animation and G. Lemaitre for help with ANOVA.

References

ALAIS, D., AND BURR, D. 2004. The ventriloquism effect results from near-optimal bimodal integration. *Current Biology 14*, 257–262.

AND, C. G. 1993. Methods for quality assessment of low bit-rate audio codecs, proceedings of the 12th aes conference. 97–107.

BERKHOUT, A., DE VRIES, D., AND VOGEL, P. 1993. Acoustic control by wave field synthesis. *J. of the Acoustical Society of America 93*, 5 (may), 2764–2778.

BLAUERT, J. 1997. *Spatial Hearing : The Psychophysics of Human Sound Localization*. M.I.T. Press, Cambridge, MA.

CHEN, J., VEEN, B. V., AND HECOX, K. 1995. A spatial feature extraction and regularization model for the head-related transfer function. *J. of the Acoustical Society of America 97* (Jan.), 439–452.

DARLINGTON, D., DAUDET, L., AND SANDLER, M. 2002. Digital audio effects in the wavelet domain. In *Proceedings of COST-G6 Conference on Digital Audio Effects, DAFX2002, Hamburg, Germany*.

2003. EBU subjective listening tests on low-bitrate audio codecs. *Technical report 3296, European Broadcast Union (EBU), Projet Group B/AIM* (june).

FOUAD, H., HAHN, J., AND BALLAS, J. 1997. Perceptually based scheduling algorithms for real-time synthesis of complex sonic environments. *proceedings of the 1997 International Conference on Auditory Display (ICAD'97),*.

GALLO, E., LEMAITRE, G., AND TSINGOS, N. 2005. Prioritizing signals for selective real-time audio processing. In *Proc. of ICAD 2005*.

HAIRSTON, W., WALLACE, M., AMD B.E. STEIN, J. V., NORRIS, J., AND SCHIRILLO, J. 2003. Visual localization ability influences cross-modal bias. *J. Cogn. Neuroscience 15*, 20–29.

HERDER, J. 1999. Optimization of sound spatialization resource management through clustering. *The Journal of Three Dimensional Images, 3D-Forum Society 13*, 3 (Sept.), 59–65.

HOCHBAUM, D. S., AND SCHMOYS, D. B. 1985. A best possible heuristic for the ık-center problem. *Mathematics of Operations Research 10*, 2 (May), 180–184.

HOWELL, D. C. 1992. *Statistical methods for psychology*. PWS-Kent.

INTERNATIONAL TELECOM. UNION. 2001-2003. Method for the subjective assessment of intermediate quality level of coding systems. *Recommendation ITU-R BS.1534-1*.

ITTI, L., KOCH, C., AND NIEBUR, E. 1998. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence 20*, 11 (Nov.), 1254–1259.

ITU-R. 1994. Methods for subjective assessment of small impairments in audio systems including multichannel sound systems. itu-r bs 1116. Tech. rep.

JOT, J.-M., AND WALSH, M. 2006. Binaural simulation of complex acoustic scenes for interactive audio. *In 121th AES Convention, San Francisco, USA. Preprint 6950*.

JOT, J.-M., LARCHER, V., AND PERNAUX, J.-M. 1999. A comparative study of 3D audio encoding and rendering techniques. *Proceedings of the AES 16th international conference, Spatial sound reproduction, Rovaniemi, Finland* (April).

KAYSER, C., PETKOV, C., LIPPERT, M., AND LOGOTHETIS, N. 2005. Mechanisms for allocating auditory attention: An auditory saliency map. *Current Biology 15* (Nov.), 1943–1947.

KELLY, M., AND TEW, A. 2002. The continuity illusion in virtual auditory space. *proc. of the 112th AES Conv., Munich, Germany* (May).

KURNIAWATI, E., ABSAR, J., GEORGE, S., LAU, C. T., AND PREMKUMAR, B. 2002. The significance of tonality index and nonlinear psychoacoustics models for masking threshold estimation. In *Proceedings of the International Conference on Virtual, Synthetic and Entertainment Audio AES22*.

LANCIANI, C. A., AND SCHAFER, R. W. 1997. Psychoacoustically-based processing of MPEG-I layer 1-2 encoded signals. In *Proc. IEEE Signal Processing Society 1997 Workshop on Multimedia Signal Processing*, 53–58.

LANCIANI, C. A., AND SCHAFER, R. W. 1999. Subband-domain filtering of MPEG audio signals. In *Proceedings of Intl. Conf. on Acoustics, Speech and Signal Processing*, 917–920.

LARCHER, V., JOT, J., GUYARD, G., AND WARUSFEL, O. 2000. Study and comparison of efficient methods for 3d audio spatialization based on linear decomposition of HRTF data. *Proc. 108th Audio Engineering Society Convention*.

LEWALD, J., EHRENSTEIN, W. H., AND GUSKI, R. 2001. Spatio-temporal constraints for auditory-visual integration. *Beh. Brain Research 121*, 1-2, 69–79.

MALHAM, D., AND MYATT, A. 1995. 3D sound spatialization using ambisonic techniques. *Computer Music Journal 19*, 4, 58–70.

MØLLER, H. 1992. Fundamentals of binaural technology. *Applied Acoustics 36*, 171–218.

PAINTER, E. M., AND SPANIAS, A. S. 2000. Perceptual coding of digital audio. *Proceedings of the IEEE 88*, 4 (Apr.).

SARLAT, L., WARUSFEL, O., AND VIAUD-DELMON, I. 2006. Ventriloquism aftereffects occur in the rear hemisphere. *Neuroscience Letters 404*, 324–329.

STOLL, G., AND KOZAMERNIK, F. 2000. EBU subjective listening tests on internet audio codecs. *EBU TECHNICAL REVIEW,* (June).

TOUIMI, A. B., EMERIT, M., AND PERNAUX, J.-M. 2004. Efficient method for multiple compressed audio streams spatialization. In *In Proceeding of ACM 3rd Intl. Conf. on Mobile and Ubiquitous multimedia*.

TOUIMI, A. B. 2000. A generic framework for filtering in subband domain. In *In Proc. of IEEE 9th Wkshp. on Digital Signal Processing, Hunt, Texas, USA*.

TSINGOS, N., GALLO, E., AND DRETTAKIS, G. 2004. Perceptual audio rendering of complex virtual environments. *Proc. SIGGRAPH'04* (August).

TSINGOS, N. 2005. Scalable perceptual mixing and filtering of audio signals using an augmented spectral representation. *Proc. of 8th Intl. Conf. on Digital Audio Effects (DAFX'05), Madrid, Spain* (Sept.).

WAND, M., AND STRASSER, W. 2004. Multi-resolution sound rendering. In *Symp. Point-Based Graphics*.

ZÖLZER, U., Ed. 2002. *DAFX - Digital Audio Effects*. Wiley.

ON-THE-FLY AUDITORY MASKING FOR SCALABLE VOIP BRIDGES

ARNAULT NAGLE[1], NICOLAS TSINGOS[2], GUILLAUME LEMAITRE[2] AND AURELIEN SOLLAUD[1]

[1] *France Telecom R&D, 2 avenue Pierre Marzin, 22307 Lannion Cedex, France*
arnault.nagle@orange-ftgroup.com
[2] *INRIA Sophia-Antipolis, 2004 route des lucioles BP 93, F-06902 Sophia-Antipolis, France*
Nicolas.Tsingos@sophia.inria.fr

Endpoints or conference servers of current audio-conferencing solutions use all the audio frames they receive in order to mix them into one final aggregate stream. However, at each time-instant, some of this content may not be audible due to auditory masking. Hence, sending corresponding frames through the network leads to a loss of bandwidth, while decoding them for mixing or spatial audio processing leads to increased processor load. In this paper, we propose a solution based on an efficient on-the-fly auditory masking evaluation. Our technique allows prioritizing audio frames in order to select only those audible for each connected client. We present results of quality tests showing the transparency of the algorithm. We describe its integration in a France Telecom audio conference server. Tests in a 3D game environment with spatialized chat capabilities show a 70% average reduction in required bandwidth, demonstrating the efficiency of our method.

1 INTRODUCTION

A significant number of VoIP systems are available, such as Microsoft Live Messenger (http://get.live.com/messenger/overview), Orange Link (http://orangelink.orange.fr/), Yahoo Messenger (http://fr.messenger.yahoo.com/), Skype (http://www.skype.com/intl/fr/), Teamspeak (http://www.goteamspeak.com/), which enable to create and manage audio chat sessions between remote participants. These systems have seen explosive growth in their usage over the last few years.

Three main configurations are usually used in VoIP audio conferences:

- centralized conferences with a mixing bridge (e.g., Orange Link) that decodes the data, generates a suitable mix for each client and streams the final result,
- loosely coupled conferences which includes multicast or multi-unicast conferences,
- semi centralized conferences with the use of a forwarding bridge. We recall the role of a forwarding bridge in Figure 1. Each client sends one stream to the server and receives from it as many streams as there are remote participants. It must next decode, potentially process (e.g., spatial audio processing) and mix them for final restitution.

In most cases, however, audio conferencing systems do not integrate spatial audio restitution. Spatialized audio gives the listener the feeling of being in a real environment where the voice of each participant is coupled with its location. This location can be arbitrarily set by the software or the user, or e.g. tied to the position of a participant in a game. Although developers can use the capabilities of the 'Xbox Live' (www.xbox.com/live/) for spatialized chat, only Unreal Tournament 2004 (http://www.unrealtournament.com/) seems to use it to date.

With the development of massively multiplayer on-line gaming, chat servers must also face the problem of dealing with an increasingly large number of simultaneous participants. This is particularly true for video game applications where an additional constraint is to send the speech data in separate streams to each participant to allow spatial audio processing prior to restitution.

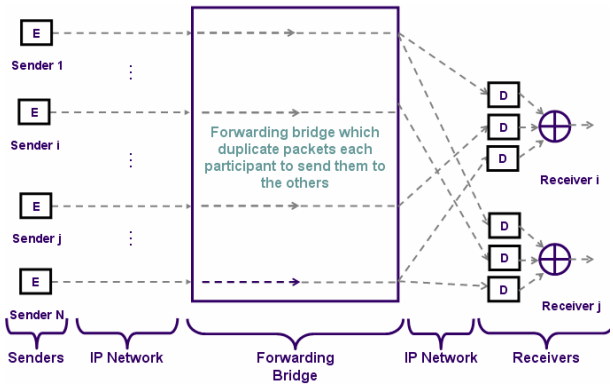


Figure 1: Illustration of the functionality of a forwarding bridge. This bridge receives a frame from a participant, duplicates and sends it towards all the other participants of the audio conference. **E**: Encoding / **D**: Decoding.

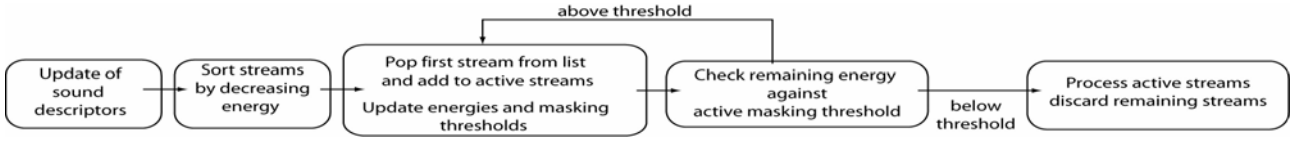


Figure 2 : Overview of our masking algorithm

In this paper, we propose a novel approach in which we stream only the audible audio frames to each participant, depending on an energy importance criteria and a small set of additional audio descriptors computed for each frame. As such, our approach is independent of the coding strategy adopted for streaming the actual audio data.
In the context of a mixing bridge, the approach does not result in any bandwidth gain, except if there is no speaker, but it optimizes the decoding of audio frames prior to mixing. For the same reasons, it is equally useful on client terminals. However, the best use of this algorithm is obviously on the forwarding bridge in order to optimize bandwidth.

This paper presents the details of our masking evaluation technique, its performance and integration into a real-time gaming application, enhanced with spatialized chat capabilities.
In section 2, we present the masking algorithm and the results of an off-line perceptual quality evaluation test. In section 3, we describe its integration in a VoIP bridge. The evaluation of this algorithm in a chat-enabled game is presented in section 4. We finally discuss our approach and outline other possible improvements of our solution before concluding.

2 EFFICIENT MASKING EVALUATION

Our masking algorithm can be decomposed in two steps. First, a set of audio descriptors must be computed for each frame of input audio signal. Typically this will be done on the client and the data will be sent to the server as additional side information together with the coded speech signal. Next, the server performs an on-the-fly masking calculation for each connected client. Since this calculation needs to consider N-1 streams for each of the N connected clients, it must be efficient in order to scale well to large numbers of participants.

2.1 Computing audio descriptors

The first stage of our masking approach computes audio descriptors from which the subsequent real-time operations can be efficiently performed.
For each frame of the input audio signal, we first compute the short-time Fourier Transform (STFT) of the audio data. For our off-line tests scenarios with 44.1 KHz signals, we used 1024 sample Hanning-windowed frames with 50 % overlap, resulting in 512 complex values in the frequency domain. Overlap is not mandatory and can be discarded to avoid additional delays in on-line applications. From the complex STFT, we then compute a number of additional descriptors:

- RMS level, including a spread-of-masking model [1, 2], for a predefined set of *i* frequency bands (e.g., typically 4 to 8 bands on an octave or Bark scale),
- Tonality T calculated as a spectral flatness measure [1]; tonality is a descriptor in *[0,1]* encoding the tonal (when close to 1) or noisy (when close to 0) nature of the signal.

The descriptors can be seen as a compact representation of the signal, typically requiring a few additional kBytes of data per second of audio signals (e.g., 3kBytes/sec. at 44.1 kHz for 1024 sample frames with 50% overlap and 8 frequency bands).
In a client-server setup, this calculation is performed by the client prior to sending the audio data and requires only minimal overhead.

2.2 Masking algorithm

From the descriptors thus obtained we can efficiently evaluate which of the input signals are going to be audible in the final mixture at a given time frame. Signals that have been identified as inaudible can be safely removed from the pipeline reducing both the arithmetic operations to perform and network traffic. Since the calculation must be carried out at each processing frame, it must be very efficient so that it does not result in significant overhead.
The masking algorithm is similar to the one presented in [3, 4] and is illustrated in Figure 2. First, all input frames are sorted according to some importance metric. In [3], a loudness metric was used but some of our recent experiments seem to indicate that the RMS level would perform equally well, if not better on average, due to specificities of some loudness models [5]. If the signals must undergo filtering or equalization operations (e.g., distance attenuation for positional audio rendering), we dynamically weight the RMS level values pre-computed for several frequency-bands to account for the influence of the filtering operations in each band. We can then compute the importance as the sum of all weighted RMS values.
Then, all signals are considered in decreasing importance order for addition to the final mixture according to the following pseudo-code:

$\mathbf{M_{mix}} = 200$
$\mathbf{P_{mix}} = 0$
$\mathbf{T} = 0$
$\mathbf{P_{toGo}} = \sum_k \mathbf{RMS_k}$
while $(dB(\mathbf{P_{toGo}}) > dB(\mathbf{P_{mix}}) - \mathbf{M}_{mix})$
and $(\mathbf{P_{toGo}} > \mathbf{ATH})$ **do**
tag signal S_k as audible
$\mathbf{P_{toGo}} -= \mathbf{RMS_k}$
$\mathbf{P_{mix}} += \mathbf{RMS_k}$
$\mathbf{T} += \mathbf{P_k} * \mathbf{T_k}$
$\mathbf{T_{mix}} = \mathbf{T}/\mathbf{P_{mix}}$
$\mathbf{M_{mix}} = 27 * \mathbf{T_{mix}} + 6 * (1 - \mathbf{T_{mix}})$
k++
end

This process basically adds the level RMS_k of each source to an estimate of the level of the final result in each band P_{mix} (initially set to zero). Accordingly, it subtracts it from an estimate of the remaining level in each band P_{toGo} (initially set to the sum of all RMS levels for all signals). The process stops when the estimated remaining level in each band is below a given threshold M_{mix} from the estimated level of the final result. The process also stops if the remaining level is below the absolute threshold of hearing *ATH* [2]. Threshold M_{mix} is adjusted according to the estimated tonality of the final result T_{mix}, following rules similar to the ones used in perceptual audio coding [1]. In our applications, a constant conservative threshold of -27 dB also gave satisfying results indicating that pre-computing and estimating tonality values might not be mandatory. Note that all operations must be performed for each frequency band, although we simplified the given pseudo-code for the sake of clarity (accordingly, all quantities should be interpreted as vectors whose dimension is the number of used frequency bands and all arithmetic operations as vector arithmetic). In particular, the process stops when the masking threshold is reached in *all* frequency bands.

2.3 Evaluation of the masking procedure

To verify the transparency of our masking evaluation, we conducted off-line quality evaluation tests to assess whether listeners aware of the algorithm principles are able to detect possible artifacts (e.g., over-masking, borderline signals rapidly switching status between masked and audible, etc.).

2.3.1 Experimental Conditions

21 subjects (17 men and 4 women) volunteered as listeners. They were aged from 23 to 40 years old with a median of 29 years old. All reported normal hearing. They all were computer scientists.
Sound reproduction was done over Sennheiser HD600 headsets connected to a laptop computer.

The test stimuli consisted of several mixtures of various sounds. The sounds were chosen in six categories: music (separated tracks of two pieces of pop music, both instrumental and vocal), speech (male and female speakers, speaking English, French, Greek, German, and Polish), environmental noises (transportation noises, animal recordings, usual office furniture noises), mixed (speech, music, noises) and elements of the reverberation of an anechoic recording of percussions computed with the image source method. This latter category is made of several delayed copies of an anechoic recording. This category was chosen because we suspected that such sounds would be difficult cases for our algorithm. In each category, we created three mixtures of sounds, made with different number of sounds. To choose which and how many sounds we mixed, we defined three levels of "masking efficiency" ("low", "medium", and "high") corresponding roughly to cases where respectively 30% to 80% of the signals were found to be masked and could be discarded in the final mixture. Each level was defined by the amount of signals actually removed by our masking algorithm. Hence, eighteen mixtures (six categories, three levels) were created. For each mixture, we created two versions: a reference mix containing all sounds, and a mix resulting from the output of the masking algorithm (i.e., a mixture of the audible sounds only). The mixtures were roughly equalized in loudness in a prior informal session.
We ran a double-blind two-alternative forced-choice (2AFC) with hidden reference procedure: subjects were presented with an interface where three buttons were displayed. The button in the middle (labeled "reference") allowed subjects to listen to the original (unprocessed) sound. The two others (labeled "A" and "B") allowed subjects to listen again to the reference (hidden reference) or to the processed mixture. The mapping between A and B and the processed/unprocessed signals was randomized at each step. Neither the listener nor the experimenter was aware of the mapping (double-blind). Subjects were instructed that one of the two signals (A or B) was different from the reference. The subject had to indicate which one they perceived as different from the reference. They could continuously switch between the three sounds, or restart each at the beginning. They were also able to define portions of the signal for looping playback. Before the test, the algorithm was explained to the subjects and they were familiarized with some clear failures of the algorithm (we had to use an older version of the algorithm to find clear impairments). This procedure was designed to get the subjects trained to identify algorithm failures and to focus on parts of the sounds where the algorithm may provide artifacts. Our hypothesis was that despite these strict conditions, subjects would be unable to correctly identify the processed sounds.

2.3.2 Analysis and discussion

Averaged over all stimuli, the identification rates per subject ranged from 39 % to 78 % with a median of 55%. Due to the binary nature of the test, it was not possible to post-screen the subject consistency hence we used the responses of all subjects as a source of variation. A t-test [6] was not able to reject the null-hypothesis: "over all the sounds, the identification rate is 50 % in the parent population" (df=18, t_{obs}=0.13, $p(t>t_{obs})>0.05$). This indicates that subjects were overall unable to identify the processed sounds better than chance.

To examine each sound individually, we performed 18 Pearson χ^2 tests with one degree of freedom [6] over the identification rates per sound (i.e. averaged over the subjects). The χ^2 hypothesis is "the identification rate is 50 %". As this test is repeated for each sound, we have to use a Bonferroni procedure for multiple comparisons [6], which amounts to decreasing the threshold of significance: $p<0.05/18$. The results of the test are represented in Figure 3.

Identification rates range from 42.8 % to 66.7 % with a median of 52.4 %. Once again, the null hypothesis of the χ^2 test could not be rejected for any of the stimuli. We obtain the same results if we consider only listeners with a musical background. It was not possible to find any relationship between identification rates and type of sound or efficiency of the masking. Thus, we can conclude that signal degradations introduced by applying our masking algorithm to compute mixtures of sounds are statistically unnoticeable.

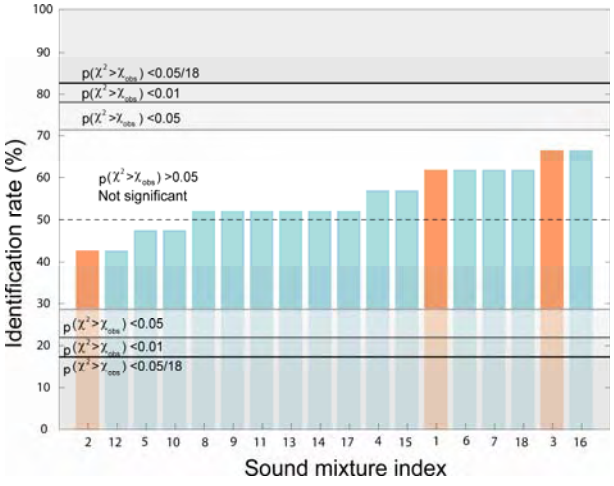


Figure 3 : Rates of identification of the processed mixtures (with masking) and results of the χ^2 test. Highlighted columns correspond to mixtures of speech signals.

However, during post-experimental interviews, several subjects reported to have found differences. Their descriptions were consistent with the expected artifacts (for instance, most of them have reported that reverberations were shorter than in the reference mixture, probably due to over-masking). Furthermore, one subject was able to precisely indicate some part of a mixture where he found a difference. Examination of the sound revealed that the algorithm had actually removed some parts of the mixture at this very location. This shows that some subjects are able to hear the differences introduced by the algorithm in some isolated cases which could not be uncovered by our test procedure. Hence, we can conclude that the masking algorithm is globally transparent, even if we can not exclude that a trained listener, listening carefully, may detect some localized artifacts.

3 INTEGRATION INTO A VOIP BRIDGE

In this section, we describe the integration of the masking algorithm into a VoIP bridge.

Figure 4 illustrates the complete audio treatment chain, from the recording of the voice of one participant P1 to the listening of the spatialized sound of another one P4. Two other participants P2 and P3 are also part of the conference. In our final gaming application, we take into consideration the spatial position of the participants in the game. This is why the Virtools game client appears on the figure. However, it is not directly related to the masking procedure which would work similarly for a non-spatialized audio output.

Each client-listener is processed as P4 and each client-speaker is processed as P1. In the following, the different steps are presented from the client and server perspectives, but temporally occur in order 1 to 5.

3.1 Client Side

3.1.1 Steps 1 and 2

The sound is recorded with the microphone of participant P1, digitized and separated into time- frames (typically 960 samples at 16kHz). For each acquired frame, the client computes necessary audio descriptors (as described in Section 2.1). To avoid additional delay, no overlap was used when analyzing the input frames. Hence, audio descriptors are computed every 60 ms. Next, frames encoded by an audio coder (we used a France Telecom codec at 32 kbits/sec equivalent to standardized G.722.1), together with their side information and the position of the user given by the Virtools client, are multiplexed into a network packet and sent to the server.

3.1.2 Step 5

At the reception end, e.g., for participant P4, audio frames are decoded from audio packets and sent to the Virtools client for final mixing and restitution.

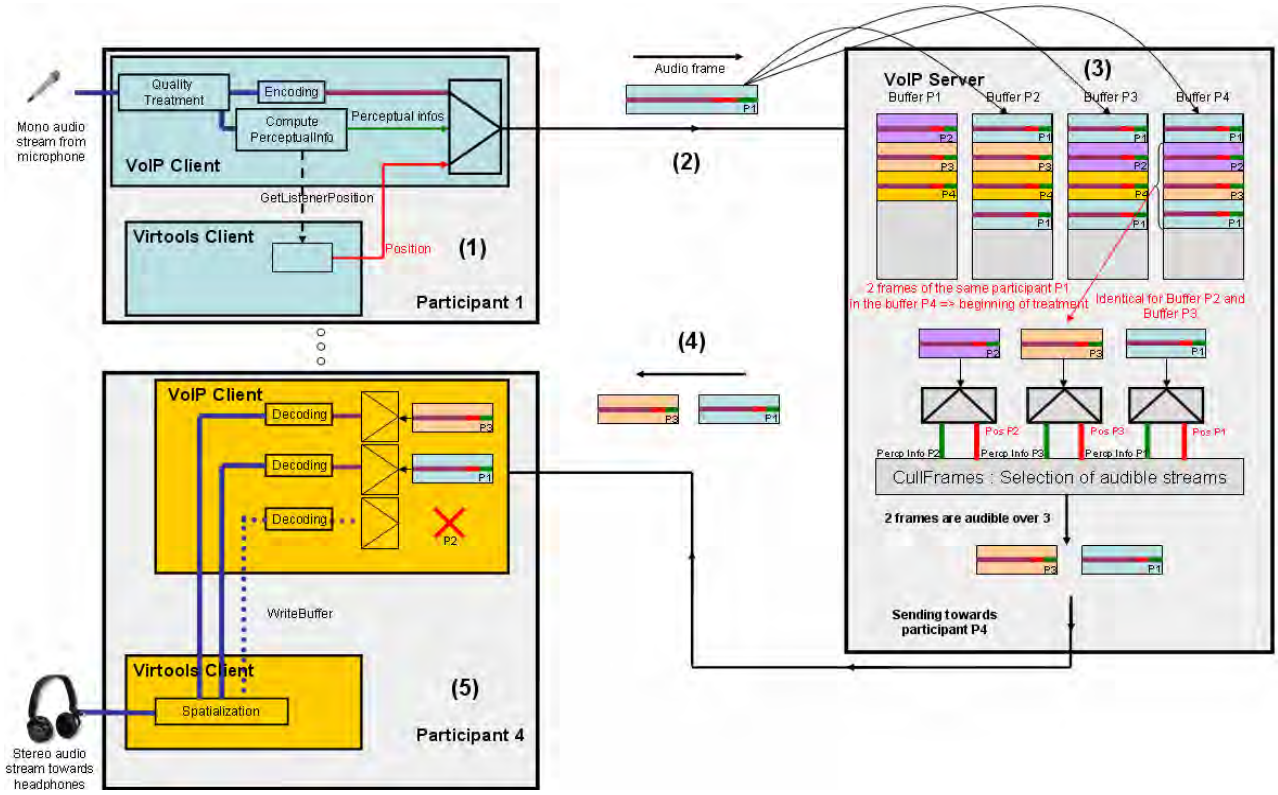


Figure 4: Overview of the VoIP /Virtools processing pipeline for a spatialized chat application.

3.2 Server Side

3.2.1 Step 3

Each audio packet (e.g., originating from P1) arrives at the VoIP server which duplicates it and inserts it in the buffers of all the others participants. On this server, each client has a buffer queue where incoming packets are stored while awaiting transfer to their final destination.

When two packets of the same participant (e.g., P1) are present in a buffer of target participant (e.g., P4), the masking calculation is launched for this buffer.

This treatment extracts perceptual descriptors of each packet available in the buffer except for the last received packet (e.g., the last one of P1). The masking algorithm, enables the selection of audible packets thanks to their associated descriptors and positions in the virtual environment (e.g., to account for distance attenuation). An order of importance is created and allows us to keep the audible packets at the output. In the situation illustrated by Figure 4, only two packets (those of P1 and P3) over three are audible for P4.

In order to avoid possible artifacts due to borderline signals rapidly alternating between audible and masked from one frame to the other, we implemented a smoothing function which remembers the two last results of the masking evaluation and decides to send the packets or not. Initially the algorithm does not send packets and it changes its state only if the current result of the algorithm and the last two are identical and opposed to the current state. A drawback of this approach is that it can erroneously discard the beginning of a sentence.

Note that a packet can be heard by some participants but not by others. Hence, each masking evaluation is performed independently for each participant.

3.2.2 Step 4

Audio packets that have been accepted by the culling function are next sent to the participant (P4 in our case).

3.3 Implementation issues

Some side effects might appear due to the recording quality on the various client platforms. In our case, tests were conducted with average consumer grade audio microphones soundcards and headphones. If no participant speaks at one time instant, the background noise can potentially be considered as a meaningful signal and will be transmitted. In fact, when there was no speaker, all the packets were sent because they all had roughly the same importance. In other cases, only packets with predominant background noise were sent. In those cases, we ideally do not want to transmit the data. However, the masking evaluation procedure does

not treat background noise differently from meaningful speech data.
We first experimented with a fixed threshold to suppress the background noise frames but it was found to depend too much on the hardware.
The implemented solution was to keep setting a very low threshold but to add a quality audio box on each sending terminal. This component located just after recording (e.g., see P1 on Figure 4) reduces noise and increases the audio level when speech is present, in order to help the masking algorithm. This noise reduction component contains:

- A high-pass filter with cut-off frequency set at 50Hz, due to the use of a wideband coder.
- A Noise Reduction block,
- And an Automatic Gain Control block to equalize the level of the audio streams of each participant.

Alternate noise removal or gating strategies could be used to solve this problem [7-10] which is common in VoIP applications and not directly related to the proposed masking evaluation.

4 EVALUATION FOR IN-GAME 3D CHAT

In order to test our algorithm in a more realistic framework, we decided to integrate the France Telecom (FT) VoIP software into the game *Flower Power Shooter* (FPS) created by the company Virtools.
The game FPS is available on "Virtools of Dassault Systèmes" website for test on http://www.virtools.com/applications/games-fps.asp. It is a multi-user game whose goal is to "shoot" the others, with paint-ball guns. Screenshots of in-game action are shown in Figure 5. Demonstration videos are available from:
http://www-sop.inria.fr/reves/OPERA/videos.
The goal was to test our masking-based optimizations for in-game spatialized chat, which is likely to create simultaneous multi-talker dialogs.

Figure 5 : Screenshots of the game Flower Power Shooter

4.1 Integration of the VoIP component in FPS

For audio conferencing applications, we saw that the best use of our masking algorithm is the case of a forwarding bridge.

Both FPS and FT VoIP software have client-server architectures. FT VoIP uses a forwarding bridge and its own signaling protocol to manage audio streams. Hence, we chose, as a preferred solution, to link the clients while keeping the two servers independent.

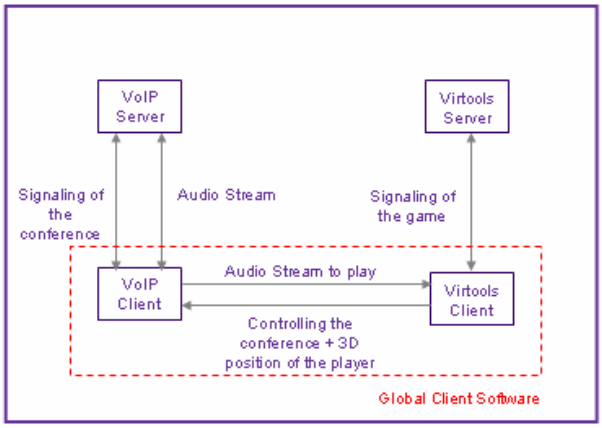


Figure 6 : Integration of the VoIP client in the Virtools client in order to provide some sound from each participant remote participant in the game

This integration leads to the control of the VoIP client by the Virtools Client, as shown in Figure 6. When a player enters the game, the Virtools server informs the other Virtools clients already present the game of his arrival. The Virtools clients signal their VoIP clients to create a new incoming stream for the new participant. The creation of the conference on the VoIP server is done at the first demand from a VoIP client. Moreover, the Virtools client continuously informs its VoIP client of the position of the user in the virtual environment.
The Virtools game client provides spatial audio rendering based on the location of each participant in the 3D game environment. The current implementation uses *openAL* hardware accelerated positional audio rendering (http://www.openal.org). Hence, upon reception, the VoIP client must feed its associated Virtools Client with audio frames received from the others participants.

4.2 Experimental setup

In order to test the algorithm, we decided to play Flower Power Shooter using several configurations. All the participants had stereo headsets with microphones (e.g., Sennheiser HMD 280 pro). One platform was used as game server and another as VoIP forwarding bridge. Pilot tests were done for 3 to 5 simultaneous participants using WIFI and Ethernet connections. For each configuration, we used two test-modes: one where nobody talks and we can observe the effect of the algorithm only in the presence of background noise, and another where players play normally while talking to each other.

We must highlight the fact that soundcards, headphones and microphones were not homogenous.

4.3 Results

Results are summarized in the following tables. “Frames normally sent” are frames which would have to be sent by a regular forwarding bridge, without our algorithm. If N participants are in the conference and one of them sends a frame, N-1 frames should be sent from the bridge to the others. Other ratios are based on this measure. “Frames accepted by the algorithm” correspond to the frames accepted by the masking algorithm but some of them can still be discarded as a result of the smoothing algorithm (see Section 3.3).

Table 1 : Results for 3 participants in the game

	In period of silence	In period of silence / talk / multi-talk
Frames normally sent	31163	169179
Frames really sent	18 ⇔ 0%	27189 ⇔ 16%
Frames accepted by the algorithm	208 ⇔ 0.6 %	37218 ⇔ 21.9%

Table 2 : Results for 4 participants in the game

	In period of silence	In period of silence / talk / multi-talk
Frames normally sent	34528	182177
Frames really sent	915 ⇔ 2.6%	30153 ⇔ 16.5%
Frames accepted by the algorithm	2025 ⇔ 5.8%	41715 ⇔ 22.8%

Table 3 : Results for 5 participants in the game

	In period of silence	In period of silence / talk / multi-talk
Frames normally sent	44161	201386
Frames really sent	1000 ⇔ 2.2%	44628 ⇔ 22.1%
Frames accepted by the algorithm	1679 ⇔ 3.8%	53831 ⇔ 26.73%

First, we can see that during total silence, the algorithm enables a significant reduction of the output bandwidth. Of course, due to the quality of the audio hardware and its associated noise, some frames can still be sent because their energy level is more important. In this case, the bridge operates as a Discontinuous Transmission System.

In periods of silence, talk or multi-talk, the reduction of output bandwidth is again quite important. We can observe the influence of the smoothing algorithm and notice that the percentage of accepted frames increases due to the augmentation of dialog possibilities. However, it does not tend to significantly grow with the number of participants.

These results depend of course of the willingness of the players to chat, of the network, of the quality of the hardware, of the smoothing function, and of the location of the avatars.

5 DISCUSSION

The use of spatialized chat in a real-time environment depends of numerous factors such as the audio hardware, the network bandwidth, the noise rejection technique, the number of results memorized in the smoothing function of the masking procedure, the level of each speaker voice and his localization etc. All of them can affect audio quality but the obtained results are very positive. Due to positional audio rendering, all participants did not have the same restitution level. For instance, the sound of a distant participant is attenuated. As a result, intelligibility can be somewhat compromised but this is a gameplay issue that will probably receive further attention by game designers as spatialized chat capabilities become more widespread. For instance, allowing players to communicate over a walkie-talkie (intelligibility preserved but no spatial aspects) or through more physical positional audio (intelligibility can be compromised by obstacles, attenuation but direction of sound is perceived) could certainly be used to drastically modify gameplay for games where teamwork is required between participants.

On the technical side, further improvements can be added to our system.

We tested a Voice Activity Detection block (VAD) piloting a Discontinuous Transmission (DTX) in the VoIP system and found it to work quite well. However, the integration of the VAD and DTX blocks into the Virtools environment created some problems; hence our results are reported without these blocks. These tools would have helped the VoIP server selecting audible audio frames and reduce the problem highlighted in Section 3.3. Furthermore, a perceptually-based noise reduction model [11] could be used instead of a simple energy-based model.

Perceptual data and positions are currently just added in audio packets without trying to further compress them to reduce the bandwidth. For instance, audio frames of 60 ms in our case are coded with 240 bytes and 3D position with 12 bytes (one 32 bit-float for each dimension). Moreover, in order to reduce the bandwidth from the server towards terminals, we could modify the audio packets with the goal of not transmitting the perceptual data and position.

Assuming a fixed bandwidth from the VoIP server to terminals, the use of variable rate coder driven by the importance factor computed by the masking function would be an interesting extension. In fact, it will allow

the optimization of the available bandwidth in the spirit of [4, 12, 13]. Streams output by a scalable variable rate coder can be cut anywhere and the audio quality depends on the length of the selected audio data. Hence, in this case, no decoding would have to be done to adapt the bandwidth, which remains consistent with the usual concept of a forwarding bridge.

In our application, we do not deal with sound effects coming from the game. Such sounds could mask or be masked by the on-line speech streams. A solution to this problem is to apply again our masking algorithm locally in each client in order to test speech signals against sound effects from the game. Our results using more general sound effects and additional masking tests presented in [3] indicate that the technique would perform equally well in this case.

Currently, the masking decision depends only on energy and tonality criteria. Others ways could certainly be explored to further prioritize the different streams, for instance the use of more perceptually oriented *saliency* metrics, as recently introduced in [14]. Further improvements could be added in the masking algorithm. For instance, we could use a binaural version of this algorithm to better account for spatial audio effects, in a way similar to [3]. However, designing a proper binaural masking model is still a challenging issue.

6 CONCLUSION

In this paper, we presented a novel approach to optimize bandwidth in the context of a voice over IP bridge for spatialized chat application. Our approach is based on an efficient on-the-fly auditory masking evaluation between all the signals generated by each participant. Masking evaluation is performed for all participants in turn so that only the audible audio frames are streamed to each client. We conducted a quality evaluation study showing that our masking algorithm yields transparent output when used to optimize mixing of a number of source signals.

In that context, our masking strategy can be used to typically remove between 20 and 80% of the original content while generating a perceptually-transparent mixture. Moreover, no delay is added by our algorithm, which is very useful in real-time application.

We integrated this algorithm in a forwarding bridge and evaluated its performance in an in-game spatialized chat context. The algorithm was found to perform very well, by discarding almost 70% of the original data in our experiments. Its performance also seems to scale well with the number of participants although we could not conduct massive chat tests during this pilot study.

We believe that such technology can be useful for massive chat applications, especially for future on-line games. Future work includes additional tests with larger numbers of participants and evaluation of a progressive streaming strategy based on the perceptual importance of each source signal.

7 ACKNOWLEDGMENTS

This work was supported by the 2-year RNTL project OPERA, co-funded by the French Ministry of Research and Ministry of Industry. More information can be found at: http://www-sop.inria.fr/reves/OPERA.

8 REFERENCES

1. Painter, J.E.M., Spanias, S., *Perceptual Coding of Digital Audio.* Proceedings of the IEEE, 2000. **88**(4).
2. Zwicker, E., Fastl, H., *Psychoacoustics: Facts and Model,* ed. Springer. 1999.
3. Tsingos, N., Gallo, E., Drettakis, G., *Perceptual Audio Rendering of Complex Virtual Environment {Proceedings of SIGGRAPH'04].* ACM Transactions on Graphics, 2004. **23**(3).
4. Tsingos, N. *Scalable Perceptual Mixing and Filtering of Audio Signals using an Augmented Spectral Representation*. in *8th Int. Conference on Digital Audio Effects (DAFx'05)*. 2005. Madrid, Spain.
5. Gallo, E., Lemaître, G., Tsingos, N. *Prioritizing audio signals for selective processing*. in *International Conference on Audio Displays*. 2005. Limerick, Ireland.
6. Howell, D.C., *Statistical methods for psychology,* ed. PWS-Kent. 1992.
7. Ephraim, Y., Malah, D. Yon, G. Camarillo, *Speech Enhancement Using a- Minimum Mean-Square Error Short-Time Spectral Amplitude Estimator.* IEEE Trans. on Acoustics, Speech and Signal, 1984. **ASSP-21**(6).
8. Mak, B., Junqua, J.-C., Reaves, B. *A robust speech/non-speech detection algorithm using time andfrequency-based features*. in *Acoustics, Speech, and Signal Processing.* 1992. San Francisco, CA, USA.
9. Renevey, P., Drygajlo, A. *Entropy Based Voice Activity Detection in Very Noisy Conditions*. in *EUROSPEECH'01*. 2001.
10. Kang, G., Lidd, M. *Automatic gain control*. in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '84.* 1984.
11. TSOUKALAS, D.E., MOURJOPOULOS, J. N., KOKKINAKIS, G., *Speech enhancement based on audible noise suppression.* IEEE transactions on speech and audio processing (IEEE trans. speech audio process.) ISSN 1063-6676 CODEN IESPEJ, 1997. **vol. 5, no6, pp. 497-514 (32 ref.)**.

12. Kelly, M.C., Tew, A.I. *The continuity illusion in virtual auditory space*. in *112th Audio Engineering Society Convention.* 2002. Munich, Germany.
13. Kelly, M.C., Tew, A.I. *The continuity illusion revisited: coding of multiple concurrent sound sources.* in *1st IEEE Benelux Workshop on Model based Processing and Coding of Audio (MPCA-2002)*. 2002. Leuven, Belgium.
14. Kayser, C., Petkov, C., Lippert, M., Logothetis, N., K., *Mechanisms for Allocating Auditory Attention: An Auditory Saliency Map, Current Biology*. Current Biology, 2005. **15**.

Chapter 6

Audio rendering from spatial recordings

In the previous chapters, we assumed that virtual sounds are emitted by a set of monophonic point sources for which a signal has to be individually generated [SHLV99, Beg94]. In the general case, source signals cannot be completely synthesized from physics principles and must be individually recorded, which requires enormous time and resources. Although this approach gives the user the freedom to control each source and freely navigate throughout the auditory scene, the overall result remains an approximation. This is due to the complexity of real-world sources, limitations of microphone pick-up patterns and limitations of the simulated sound propagation models. On the opposite end of the spectrum, spatial sound recording techniques which encode directional components of the soundfield [Mer02, ME04b, Sou, Strb] can be directly used to acquire and playback real-world auditory environments as a whole. They produce realistic results but offer little control, if any, at the playback end. In particular, they are acquired from a single location in space and only encode directional information, which makes them insufficient for free-walkthrough applications or rendering of large near-field sources. In such spatially-extended cases, correct reproduction requires sampling the soundfield at several locations and encoding the 3D position and not only the incoming direction of the sounds. In practice, the use of such singlepoint recordings is mostly limited to the rendering of an overall surround ambiance that can possibly be rotated around the listener.

This final chapter explores an approach to build and render virtual auditory scenes directly from non-coincident spatial recordings. This approach combines an *off-line analysis* of real-world multi-point recordings with an *interactive post-processing* allowing for flexible 3D navigation and editing of the original recording. As the complexity of the environments to author and render increases, this is an attractive alternative to modeling auditory scenes with collection of point sources.

More details can be found in the related publications:

- Emmanuel Gallo, Nicolas Tsingos and Guillaume Lemaitre.
  3D-Audio Matting, Post-editing and Re-rendering from Field Recordings.
  EURASIP JASP, special issue on Spatial Sound and Virtual Acoustics, 2007.

- Emmanuel Gallo and Nicolas Tsingos.
  Extracting and Re-rendering Structured Auditory Scenes from Field Recordings.
  Proceedings of the 30th AES Intl. Conf. on Intelligent Audio Environments, 2007.

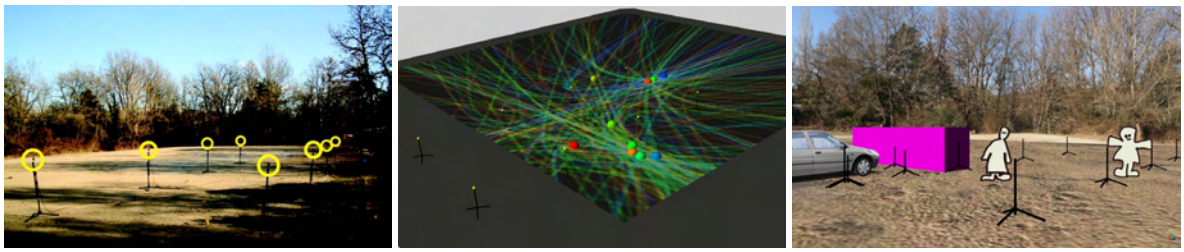


Figure 6.1: Left: We use multiple arbitrarily positioned microphones (circled in yellow) to simultaneously record real-world auditory environments. Middle: We analyze the recordings to extract the positions of various sound components through time. Right: This high-level representation allows for post-editing and re-rendering the acquired soundscape within generic 3D-audio rendering architectures

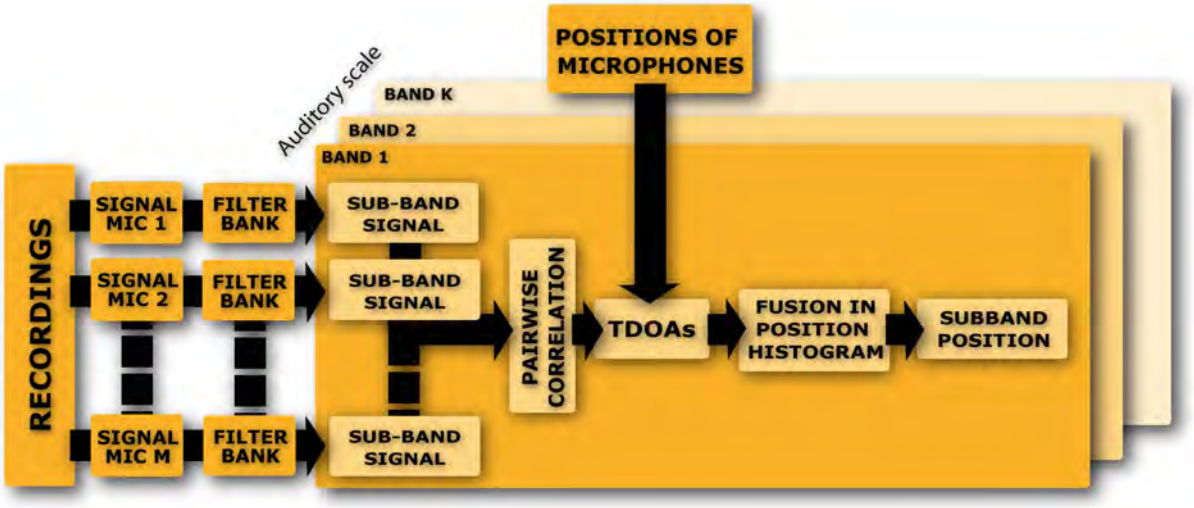


Figure 6.2: Overview of the analysis algorithm used to construct a spatial mapping for the acquired soundscapes.

6.1 Extracting auditory scene structure from recordings

We propose to record a real-world soundscape using arbitrarily placed omnidirectional microphones in order to get a good acoustic sampling from a variety of locations within the environment. Contrary to most related approaches, we use widely-spaced microphone arrays (Figure 6.1). Any studio microphones can be used for this purpose, which makes the approach well suited to production environments. We also propose an image-based calibration strategy making the approach practical for field applications. The obtained set of recordings is analyzed in an off-line pre-processing step in order to segment various auditory components and associate them with the position in space from which they were emitted. To compute this spatial mapping, we split the signal into short time-frames and a set of frequency subbands. We then use classical time-difference of arrival techniques between all pairs of microphones to retrieve a position for each subband at each time-frame. An overview of the analysis process is shown in Figure 6.2.

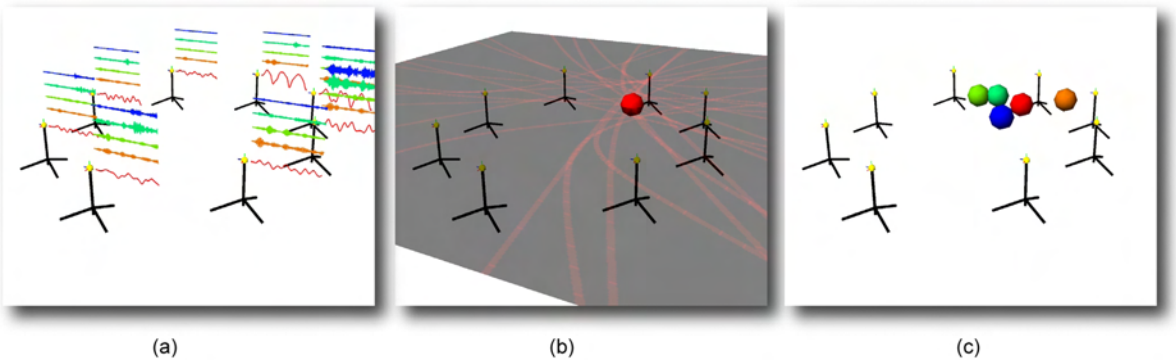


Figure 6.3: Illustration of the construction of the global spatial mapping for the captured sound-field. (a) At each time-frame, we split the signals recorded by each microphone into the same set of frequency subbands. (b) Based on time-difference of arrival estimation between all pairs of recordings, we sample all corresponding hyperbolic loci to obtain a position estimate for the considered subband. (c) Position estimates for all subbands at the considered time-frame (shown as colored spheres).

6.1.1 Localizing sound events

We acquire real-world soundscapes using a number of omnidirectional microphones and a multi-channel recording interface connected to a laptop computer.

In order to extract correct spatial information from the recordings, it is necessary to first retrieve the 3D locations of the microphones. We use REALVIZ *ImageModeler* (www.realviz.com) to extract the 3D locations from a small set of photographs (4 to 8 in our test examples) taken from several angles, but any standard algorithm can be applied for this step [Fau93].

We then analyze the recordings in order to produce a high-level representation of the captured soundscape. This high-level representation is a mapping, global to the scene, between different frequency subbands of the recordings and positions in space from which they were emitted (Figure 6.3). We consider each frequency subband as a unique point source for which a single position has to be determined. We chose to use a Time-difference of arrival (TDOA) strategy to determine the location of the various auditory events. Analysis of the recordings is done on a frame by frame basis using short time-windows (typically 20ms long or 1024 samples at CD quality). For a given source position and a given pair of microphones, the propagation delay from the source to the microphones generates a measurable time-difference of arrival. The set of points which generate the same TDOA defines an hyperboloid surface in 3D (or an hyperbola in 2D) which foci are the locations of the two microphones (Figure 6.3 (b)).

6.1.2 Background and foreground segmentation

We further segment stationary background noise from non-stationary sound events using the technique by Ephraim and Malah [EM84], originally developed for denoising of speech signals. This approach assumes that the distributions of Fourier coefficients for signal and noise are statistically independent zero-mean Gaussian random variables. Under this assumption, the spectral amplitude of the denoised signal is estimated using a minimum mean-square error criterion. The background noise signal is then simply obtained by subtracting the denoised signal from the original. The separated foreground and background components are both processed using the analysis pipeline described in Section 6.1.1 (see also Figure 6.3). However, in the case of the background component, we obtain noisier position estimates since this component will generally correspond to background noise and sources with low signal-to-noise

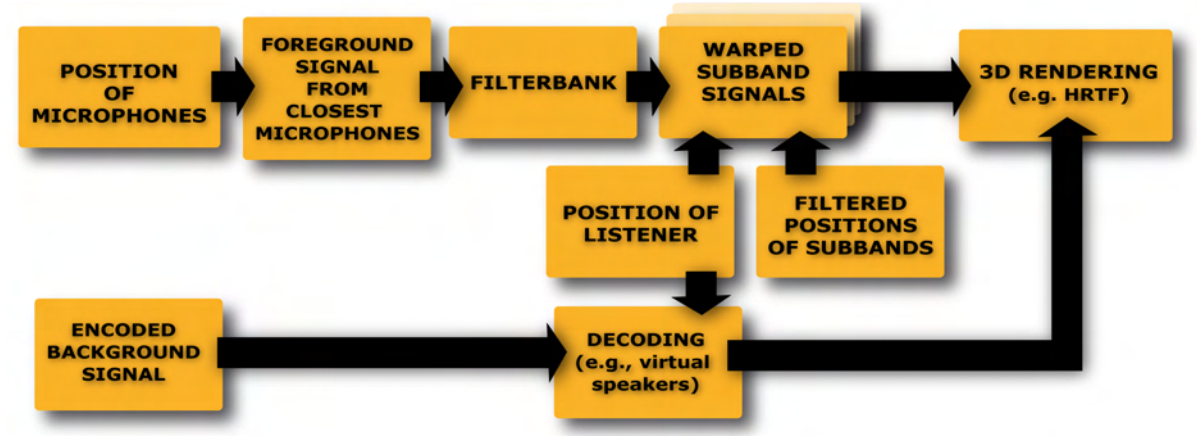


Figure 6.4: Overview of our re-synthesis pipeline. Foreground sound events are rendered as point sources while background sounds are encoded using a low-order spherical harmonics decomposition.

ratios. In order to produce a smooth spatial background texture, we use the obtained positions to encode the corresponding subband signals on a 1st-order spherical harmonic basis. (Figure 6.4).

6.2 Post-editing and re-rendering the scene

Our approach can lead to spatial audio coding applications for live audio footage in a way similar to [Pul06, GJ06, BHF+05], but it also offers novel decoding/authoring capabilities not available with previous techniques. Since it uses a world-space representation of spatial auditory cues, it allows for flexible re-rendering of the acquired scene, independent of the reproduction setup (headphones, multi-channel, etc.).

6.2.1 Free-viewpoint rendering

Our approach allows for a free-viewpoint spatial audio rendering of the acquired soundscapes. As the virtual listener moves throughout the scene, the foreground component is rendered using a collection of point sources corresponding to each time-frequency atom.

At run-time during an interactive simulation, we use the previously computed spatial mapping to properly warp the original recordings when the virtual listener moves throughout the environment. With an additional clustering step, we recombine frequency subbands emitted from neighboring locations and segment spatially-consistent sound events. This allows us to select and post-edit subsets of the acquired auditory environment. Finally the location of the clusters is used for spatial audio restitution within standard 3D-audio APIs.

The background component is simply rotated based on the current orientation of the listener in order to provide a consistent rendering. No warping is applied to the background component in this case.

Figure 6.4 shows our complete re-rendering pipeline. Example acquisition setups and reconstructed virtual scenes are shown in Figures 6.1 and 6.5.

6.2.2 Content editing

We can also alter the contents of the recordings in several ways.

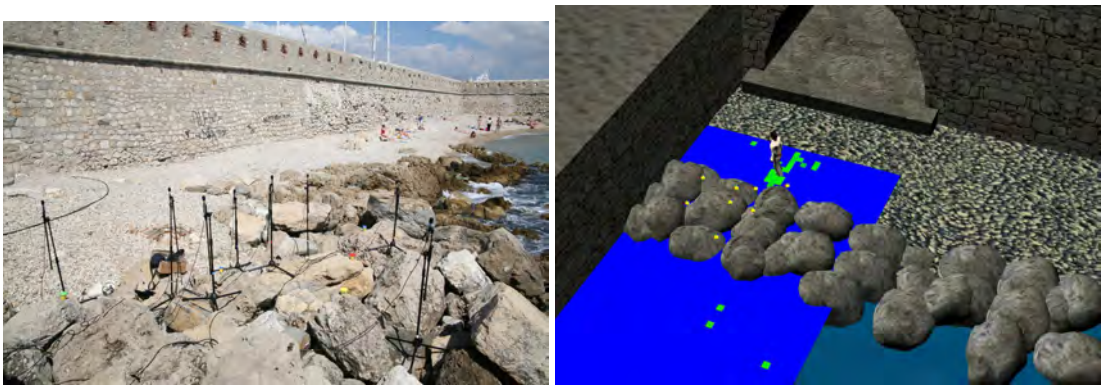


Figure 6.5: Left:Recording setup used for the seashore recordings. Right: Example virtual reconstruction of a seashore with walking pedestrian. Yellow spheres correspond to the locations of the microphones used for recording.

Source re-localization and modification

Using our technique, we can selectively choose and modify various elements of the original recordings. For instance, we can select any spatial area in the scene and simply relocate all sources included in the selected region.

Our two-layer model allows for independent control of the background and foreground components. Their overall level can be adjusted globally or locally, for instance to attenuate foreground sounds with local virtual occluders while preserving the background. The foreground events can also be copied and pasted over a new background ambiance.

Compositing and Real/Virtual integration

Since our recording setups are spatially calibrated, we can integrate several environments into a single composite rendering which preserves the relative size and positioning of the various sound sources. Future work might include merging the representations in order to limit the number of composite recordings (for instance by "re-projecting" the recordings of one environment into the recording setup of the other and mixing the resulting signals).

Our approach permits spatially consistent compositing of virtual sources within real-world recordings. We can also integrate virtual objects, such as walls, and make them interact with the original recordings. For instance, by performing real-time ray-casting between the listener and the location of the frequency subbands, we can add occlusion effects due to a virtual obstacle using a model similar to [TG98]. Of course, perfect integration would also require correcting for the reverberation effects between the different environments to composite. Currently, we experimented only in environments with limited reverberation but blind extraction of reverberation parameters [BW02] and blind deconvolution are complementary areas of future research in order to better composite real and virtual sound-fields.

6.3 Discussion

We presented an approach to convert field recordings into a structured representation suitable for generic 3D audio processing and integration with 2D or 3D visual content. It applies both to outdoor environments or indoor environments with limited reverberation, provides a compact encoding of the spatial

auditory cues and captures propagation and reverberation effects which would be very difficult to render with the same level of realism using traditional virtual acoustics simulations.

In the future, we would like to improve on our background/foreground segmentation approach, possibly based on auditory *saliency* models [KPLL05] or taking advantage of the signals from all microphones. Alternative sparse representations of the signals [MZ93, LS00] could also be explored in order to improve our approach.

Perceptual comparisons with reference binaural and B-format recordings showed that our approach outperforms B-format recordings and can get close to reference binaural recordings when all time-frequency atoms are rendered as foreground point sources. However, artefacts due to background noise lead to reduced signal quality. An alternative solution was proposed based on the explicit segmentation of stationary "background noise" and non-stationary "foreground events". While the signal quality is significantly improved when re-rendering, spatial cues were perceived to be degraded, probably due to non-optimal background separation. Further comparisons to other sound-field acquisition techniques, for instance based on high-order spherical harmonic encoding [AW02, ME04a], Fourier-Bessel decomposition [LBM03, LBM04] or directional audio coding [Pul06, PF06] would also be of primary interest to evaluate the quality vs. flexibility/applicability tradeoffs of the various approaches.

3D-Audio Matting, Post-editing and Re-rendering from Field Recordings

Emmanuel Gallo[1,2], Nicolas Tsingos[1] and Guillaume Lemaitre[1]
[1]REVES/INRIA and [2]CSTB,
Sophia-Antipolis, France*

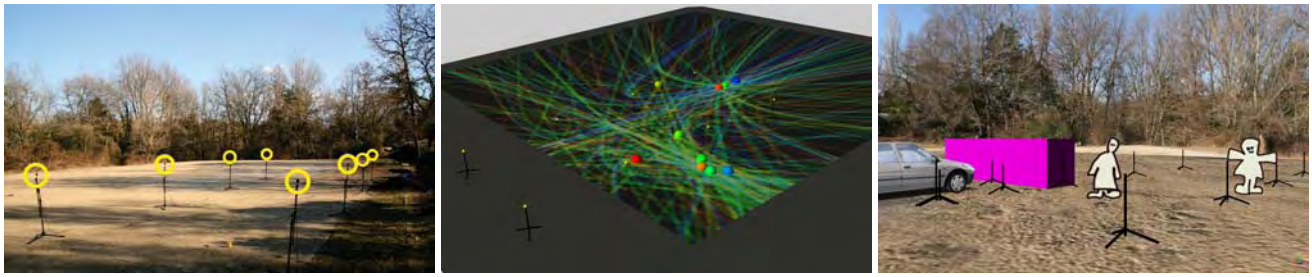


Figure 1: Left: We use multiple arbitrarily positioned microphones (circled in yellow) to simultaneously record real-world auditory environments. Middle: We analyze the recordings to extract the positions of various sound components through time. Right: This high-level representation allows for post-editing and re-rendering the acquired soundscape within generic 3D-audio rendering architectures.

Abstract

We present a novel approach to real-time spatial rendering of realistic auditory environments and sound sources recorded live, in the field. Using a set of standard microphones distributed throughout a real-world environment we record the sound-field simultaneously from several locations. After spatial calibration, we segment from this set of recordings a number of auditory components, together with their location. We compare existing time-delay of arrival estimations techniques between pairs of widely-spaced microphones and introduce a novel efficient hierarchical localization algorithm. Using the high-level representation thus obtained, we can edit and re-render the acquired auditory scene over a variety of listening setups. In particular, we can move or alter the different sound sources and arbitrarily choose the listening position. We can also composite elements of different scenes together in a spatially consistent way. Our approach provides efficient rendering of complex soundscapes which would be challenging to model using discrete point sources and traditional virtual acoustics techniques. We demonstrate a wide range of possible applications for games, virtual and augmented reality and audio-visual post-production.

Keywords: Virtual Environments, Spatialized Sound, Audio Recording Techniques, Auditory Scene Analysis, Image-based rendering, Matting and compositing

*{Emmanuel.Gallo|Nicolas.Tsingos}@sophia.inria.fr
http://www-sop.inria.fr/reves/
Guillaume Lemaitre is now with IRCAM.

1 Introduction

While hardware capabilities allow for real-time rendering of increasingly complex environments, authoring realistic virtual audio-visual worlds is still a challenging task. This is particularly true for interactive spatial auditory scenes for which few content creation tools are available.

Current models for authoring interactive 3D-audio scenes often assume that sound is emitted by a set of monophonic point sources for which a signal has to be individually generated. In the general case, source signals cannot be completely synthesized from physics-based models and must be individually recorded, which requires enormous time and resources. Although this approach gives the user the freedom to control each source and freely navigate throughout the auditory scene, the overall result remains an approximation due to the complexity of real-world sources, limitations of microphone pick-up patterns and limitations of the simulated sound propagation models.

On the opposite end of the spectrum, spatial sound recordings which encode directional components of the sound-field can be directly used to acquire live auditory environments as a whole [44, 66]. They produce lifelike results but offer little control, if any, at the playback end. In particular, they are acquired from a single location in space, which makes them insufficient for walkthrough applications or rendering of large near-field sources. In practice, their use is mostly limited to the rendering of an overall ambiance. Besides, since no explicit position information is directly available for the sound sources, it is difficult to tightly couple such spatial recordings with matching visuals.

This paper presents a novel analysis-synthesis approach which bridges the two previous strategies. Our method builds a higher-level spatial description of the auditory scene from a set of field recordings (Figure 1). By analyzing how different frequency components of the recordings reach the various microphones through time, it extracts both spatial information and audio content for the most significant sound events present in the acquired environment. This spatial mapping of the auditory scene can then be used for post-processing and re-rendering the original recordings. Re-rendering is achieved through a frequency-dependent warping of the recordings, based on the estimated positions of several frequency subbands of the signal. Our approach makes positional

information about the sound sources directly available for generic 3D-audio processing and integration with 2D or 3D visual content. It also provides a compact encoding of complex live auditory environments and captures complex propagation and reverberation effects which would be very difficult to render with the same level of realism using traditional virtual acoustics simulations.

Our work complements image-based modeling and rendering approaches in computer graphics [16, 28, 12, 5]. Moreover, similar to the *matting* and *compositing* techniques widely used in visual effects production [54], we show that the various auditory components segmented out by our approach can be pasted together to create novel and spatially consistent soundscapes. For instance, foreground sounds can be integrated in a different background ambiance.

Our technique opens many interesting possibilities for interactive 3D applications such as games, virtual/augmented reality or off-line post-production. We demonstrate its applicability to a variety of situations using different microphone setups.

2 Related work

Our approach builds upon prior work in several domains including spatial audio acquisition and restitution, structure extraction from audio recordings and blind source separation. A fundamental difference between the approaches is whether they attempt to capture the spatial structure of the wavefield through mathematical or physical models or attempt to perform a higher-level auditory scene analysis to retrieve the various, perceptually meaningful, sub-components of the scene and their 3D location. The following sections give a short overview of the background most relevant to our problem.

2.1 Spatial sound-field acquisition and restitution

Processing and compositing live multi-track recordings is of course a widely used method in motion-picture audio production [73]. For instance, recording a scene from different angles with different microphones allows the sound editor to render different audio perspectives, as required by the visual action. Thus, producing synchronized sound-effects for films requires carefully planned microphone placement so that the resulting audio track perfectly matches the visual action. This is especially true since the required audio material might be recorded at different times and places, before, during and after the actual shooting of the action on stage. Usually, simultaneous monaural or stereophonic recordings of the scene are composited by hand by the sound designer or editor to yield the desired track, limiting this approach to off-line post-production. Surround recording setups (e.g., *Surround Decca Trees*) [67, 68], which historically evolved from stereo recording, can also be used for acquiring a sound-field suitable for restitution in typical cinema-like setups (e.g., 5.1-surround). However, such recordings can only be played-back directly and do not support spatial post-editing.

Other approaches, more physically and mathematically grounded, decompose the wavefield incident on the recording location on a basis of spatial harmonic functions such as spherical/cylindrical harmonics (e.g., *Ambisonics*) [25, 44, 18, 38, 46] or generalized Fourier-Bessel functions [36]. Such representations can be further manipulated and decoded over a variety of listening setups. For instance, they can be easily rotated in 3D space to follow the listener's head orientation and have been successfully used in immersive virtual reality applications. They also allow for beamforming applications, where sounds emanating from any specified direction can be further isolated and manipulated. However, these techniques are practical mostly for low order decompositions (order 2 already requiring 9 audio channels) and, in return, suffer from limited directional accuracy [31]. Most of them also require specific microphones [2, 48, 66, 37] which are not widely available and whose bandwidth usually drops when the spatial resolution increases. Hence, higher-order microphones do not usually deliver production-grade audio quality, maybe with the exception of *Trinnov's SRP* system [37] (www.trinnov.com) which uses regular studio microphones but is dedicated to 5.1-surround restitution. Finally, a common limitation of these approaches is that they use coincident recordings which are not suited to rendering walkthroughs in larger environments.

Closely related to the previous approach is wave-field synthesis/holophony [9, 10]. Holophony uses the Fresnel-Kirchoff integral representation to sample the sound-field inside a region of space. Holophony could be used to acquire live environments but would require a large number of microphones to avoid aliasing problems, which would jeopardize proper localization of the reproduced sources. In practice, this approach can only capture a live auditory scene through small acoustic "windows". In contrast, while not providing a physically-accurate reconstruction of the sound-field, our approach can provide stable localization cues regardless of the frequency and number of microphones.

Finally, some authors, inspired from work in computer graphics and vision, proposed a dense sampling and interpolation of the *plenacoustic function* [3, 20] in the manner of *lumigraphs* [26, 39, 12, 5]. However, these approaches remain mostly theoretical due to the required spatial density of recordings. Such interpolation approaches have also been applied to measurement and rendering of reverberation filters [53, 27]. Our approach follows the idea of acquiring the plenacoustic function using only a sparse sampling and then warping between this samples interactively, e.g., during a walkthrough. In this sense, it could be seen as an "unstructured plenacoustic rendering".

2.2 High-level auditory scene analysis

A second large family of approaches aims at identifying and manipulating the components of the sound-field at a higher-level by performing auditory scene analysis [11]. This usually involves extracting spatial information about the sound sources and segmenting out their respective content.

Spatial feature extraction and restitution

Some approaches extract spatial features such as binaural cues (interaural time-difference, interaural level difference, interaural correlation) in several frequency subbands of stereo or surround recordings. A major application of these techniques is efficient multi-channel audio compression [8, 23] by applying the previously extracted binaural cues to a monophonic down-mix of the original content. However, extracting binaural cues from recordings requires an implicit knowledge of the restitution system.

Similar principles have also been applied to flexible rendering of directional reverberation effects [47] and analysis of room responses [46] by extracting direction of arrival information from coincident or near-coincident microphone arrays [55].

This paper generalizes these approaches to multi-channel field recordings using arbitrary microphone setups and no *a priori* knowledge of the restitution system. We propose a direct extraction of the 3D position of the sound sources rather than binaural cues or direction of arrival.

Blind source separation

Another large area of related research is blind source separation (BSS) which aims at separating the various sources from one or several mixtures under various mixing models [71, 52]. Most recent BSS approaches rely on a sparse signal representation in some space of basis functions which minimizes the probability that a

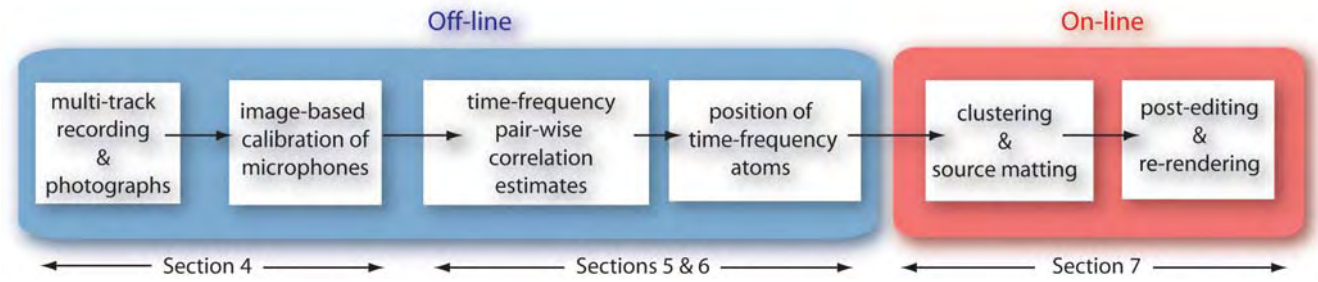


Figure 2: Overview of our pipeline. In an off-line phase, we first analyze multi-track recordings of a real-world environment to extract the location of various frequency subcomponents through time. At run-time, we aggregate these estimates into a target number of clustered sound sources for which we reconstruct a corresponding signal. These sources can then be freely post-edited and re-rendered.

high-energy coefficient at any time-instant belongs to more than one source [58]. Some work has shown that such sparse coding does exists at the cortex level for sensory coding [41]. Several techniques have been proposed such as independent component analysis (ICA) [17, 63] or the more recent *DUET* technique [32, 74] which can extract several sources from a stereophonic signal by building an inter-channel delay/amplitude histogram in Fourier frequency domain. In this aspect, it closely resembles the aforementioned binaural cue coding approach. However, most BSS approaches do not separate sources based on spatial cues, but directly solve for the different source signals assuming *a priori* mixing models which are often simple. Our context would be very challenging for such techniques which might require knowing the number of sources to extract in advance, or need more sensors than sources in order to explicitly separate the desired signals. In practice, most auditory BSS techniques are devoted to separation of speech signals for telecommunication applications but other audio applications include upmixing from stereo to 5.1 surround formats [6].

In this work, however, our primary goal is not to finely segment each source present in the recorded mixtures but rather to extract enough spatial information so that we can modify and re-render the acquired environment while preserving most of its original content. Closer in spirit, the DUET technique has also been used for audio interpolation [57]. Using a pair of closely spaced microphones, the authors apply DUET to re-render the scene at arbitrary locations along the line passing through the microphones. The present work extends this approach to arbitrary microphone arrays and rerendering at any 3D location in space.

3 Overview

We present a novel acquisition and 3D-audio rendering pipeline for modeling and processing realistic virtual auditory environments from real-world recordings.

We propose to record a real-world soundscape using arbitrarily placed omnidirectional microphones in order to get a good acoustic sampling from a variety of locations within the environment. Contrary to most related approaches, we use widely-spaced microphone arrays. Any studio microphones can be used for this purpose, which makes the approach well suited to production environments. We also propose an image-based calibration strategy making the approach practical for field applications. The obtained set of recordings is analyzed in an off-line pre-processing step in order to segment various auditory components and associate them with the position in space from which they were emitted. To compute this spatial mapping, we split the signal into short time-frames and a set of frequency subbands. We then use classical time-difference of arrival techniques between all pairs of microphones to retrieve a position for each subband at each time-frame. We evaluate the performance of existing approaches in our context and present an improved hierarchical source localization technique from the obtained time-differences.

This high-level representation allows for flexible and efficient on-line re-rendering of the acquired scene, independent of the restitution system. At run-time during an interactive simulation, we use the previously computed spatial mapping to properly warp the original recordings when the virtual listener moves throughout the environment. With an additional clustering step, we recombine frequency subbands emitted from neighboring locations and segment spatially-consistent sound events. This allows us to select and post-edit subsets of the acquired auditory environment. Finally the location of the clusters is used for spatial audio restitution within standard 3D-audio APIs.

Figure 2 shows an overview of our pipeline. Sections 4, 5 and 6 describe our acquisition and spatial analysis phase in more detail. Section 7 presents the on-line spatial audio resynthesis based on the previously obtained spatial mapping of the auditory scene. Finally, Section 8 describes several applications of our approach to realistic rendering, post-editing and compositing of real-world soundscapes.

4 Recording setup and calibration

We acquire real-world soundscapes using a number of omnidirectional microphones and a multi-channel recording interface connected to a laptop computer. In our examples, we used up to 8 identical *AudioTechnica AT3032* microphones and a *Presonus Firepod* firewire interface running on batteries. The microphones can be arbitrarily positioned in the environment. Section 8 shows various possible setups. To produce the best results, the microphones should be placed so as to provide a compromise between the signal-to-noise ratio of the significant sources and spatial coverage.

In order to extract correct spatial information from the recordings, it is necessary to first retrieve the 3D locations of the microphones. Maximum-likelihood autocalibration methods could be used based on the existence of pre-defined source signals in the scene [50], for which the time-of-arrival (TOA) to each microphone has to be determined. However, it is not always possible to introduce calibration signals at a proper level in the environment. Hence, in noisy environments obtaining the required TOAs might be difficult, if not impossible. Rather, we use an image-based technique from photographs which ensures fast and convenient acquisition on location, not requiring any physical measurements or homing device. Moreover, since it is not based on acoustic measurements, it is not subject to background noise and is likely to produce better results. We use REALVIZ *ImageModeler* (www.realviz.com) to extract the 3D locations from a small set of photographs (4 to 8 in our test examples) taken from several angles, but any standard algorithm can be applied for this step [24]. To facilitate the process we

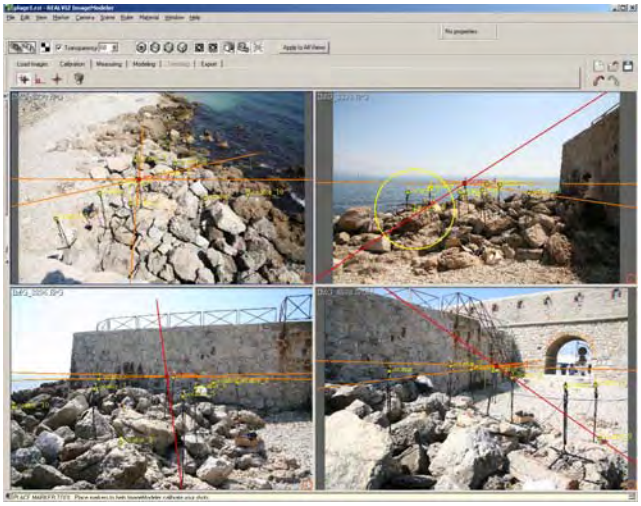


Figure 3: We retrieve the position of the microphones from several photographs of the setup using a commercial image-based modeling tool. In this picture, we show four views of a recording setup, position of the markers and the triangulation process yielding the locations of the microphone capsules.

place colored markers (tape or balls of modeling clay) on the microphones, as close as possible to the actual location of the capsule, and on the microphone stands. Additional markers can also be placed throughout the environment to obtain more input data for calibration. The only constraint is to provide a number of non-coplanar calibration points to avoid degenerate cases in the process. In our test examples, the accuracy of the obtained microphone locations was of the order of one centimeter. Image-based calibration of the recording setup is a key aspect of our approach since it allows for treating complex field recording situations such as the one depicted in Figure 3 where microphones stands are placed on large irregular rocks on a seashore.

5 Propagation model and assumptions for source matting

From the M recorded signals, our final goal is to localize and re-render a number J of *representative sources* which offer a good perceptual reconstruction of the original soundscape captured by the microphone array. Our approach is based on two main assumptions.

First, we consider that the recorded sources can be represented as point emitters and assume an ideal anechoic propagation model. In this case, the mixture $x_m(t)$ of N sources $s_1(t),..,s_n(t)$ recorded by the m^{th} microphone can be expressed as:

$$x_m(t) = \sum_{n=1}^{N} a_{mn}(t)s_n(t-\delta_{mn}(t)), \tag{1}$$

where parameters $a_{mn}(t)$ and $\delta_{mn}(t)$ are the attenuation coefficients and time-delays associated with the n^{th} source and the m^{th} microphone.

Second, since our environments contain more than one active source simultaneously, we consider K frequency subbands, $K >= J$, as the basic components we wish to position in space at each time-frame (Figure 5 (a)). We choose to use non-overlapping frequency subbands uniformly defined on a Bark scale [49] to provide a more psycho-acoustically relevant subdivision of the audible spectrum (in our examples, we experimented with 1 to 32 subbands).

In frequency domain, the signal x_m filtered in the k^{th} Bark band can be expressed at each time-frame as:

$$Y_{km}(z) = W_k(z)\sum_{t=1}^{T} x_m(t)e^{-j(2\pi zt/T)} = W_k(z)X_m(z), \tag{2}$$

where

$$W_k(f) = \begin{cases} 1 & \frac{25k}{K} < Bark(f) < \frac{25(k+1)}{K} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

$$Bark(f) = 13atan(\frac{0.76f}{1000}) + 3.5atan(\frac{f^2}{7500^2}), \tag{4}$$

$f = z/Zf_s$ is the frequency in Hertz, f_s is the sampling rate and $X_m(z)$ is the $2Z$-point Fourier transform of $x_m(t)$. We typically record our live signals using 24-bit quantization and $f_s = 44.1KHz$. The subband signals are computed using $Z = 512$ with a Hanning window and 50% overlap before storing them back into time-domain for later use.

At each time-frame, we construct a new representation for the captured soundfield at an arbitrary listening point as:

$$\hat{x}(t) \approx \sum_{j=1}^{J}\sum_{k=1}^{K} \hat{\alpha}^j_{km} y_{km}(t+\hat{\delta}_{km}), \forall m \tag{5}$$

where $y_{km}(t)$ is the inverse Fourier transform of $Y_{km}(z)$, $\hat{\alpha}^j_{km}$ and $\hat{\delta}_{km}$ are correction terms for attenuation and time-delay derived from the estimated positions of the different subbands. The term $\hat{\alpha}^j_{km}$ also includes a matting coefficient representing how much energy within each frequency subband should belong to to each representative source. In this sense, it shares some similarity with the *time-frequency masking* approach of [74].

The obtained representation can be made to match the acquired environment if $K >= N$ and if, following a sparse coding hypothesis, we further assume that the contents of each frequency subband belong to a single source at each time-frame. This hypothesis is usually referred to as *W-disjoint orthogonality* [74] and given N sources $S_1,..,S_N$ in Fourier domain, it can be expressed as:

$$S_i(z)S_j(z) = 0 \;\; \forall i \neq j \tag{6}$$

When the two previous conditions are not satisfied, the representative sources will correspond to a mixture of the original sources and Equ. 5 will lead to a less accurate approximation.

6 Spatial mapping of the auditory scene

In this step of our pipeline, we analyze the recordings in order to produce a high-level representation of the captured soundscape. This high-level representation is a mapping, global to the scene, between different frequency subbands of the recordings and positions in space from which they were emitted (Figure 5).

Following our previous assumptions, we consider each frequency subband as a unique point source for which a single position has to be determined. Localization of a sound source from a set of audio recordings, using a single-propagation-path model, is a well studied problem with major applications in robotics, people tracking and sensing, teleconferencing (e.g, automatic camera steering) and defense. Approaches rely either on time-difference of arrival (TDOA) estimates [1, 34, 30], high-resolution spectral estimation (e.g., MUSIC) [64, 35] or steered response power using a

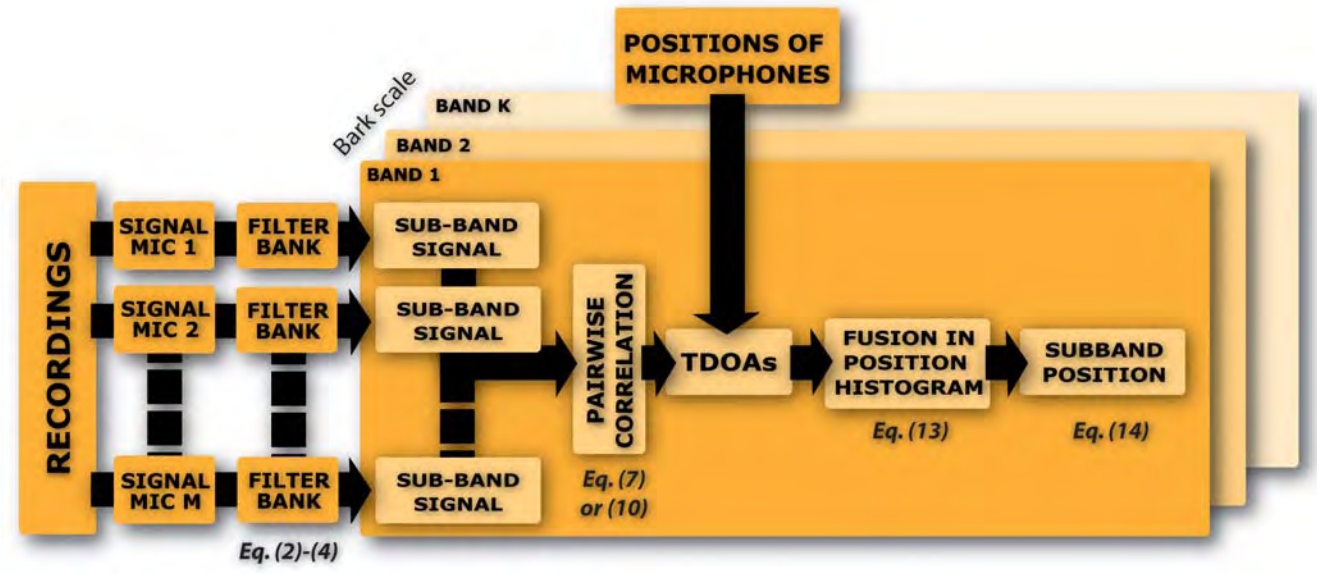


Figure 4: Overview of the analysis algorithm used to construct a spatial mapping for the acquired soundscapes.

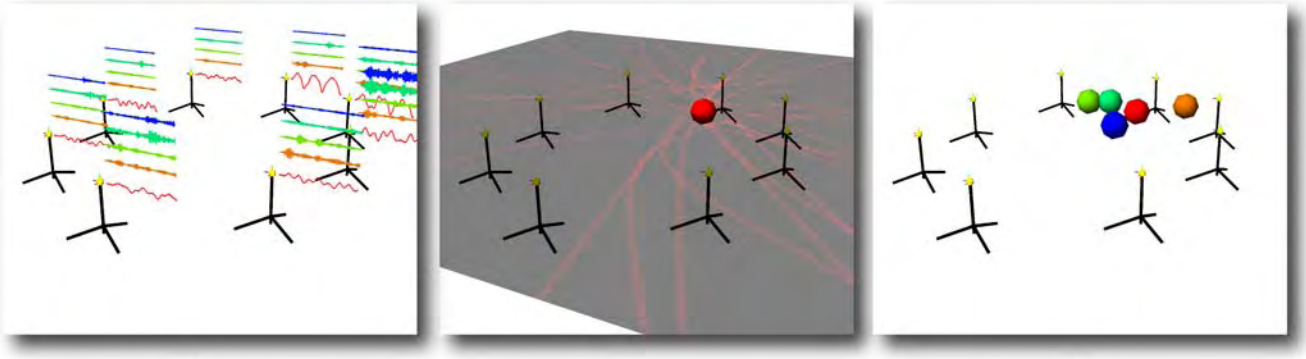


Figure 5: Illustration of the construction of the global spatial mapping for the captured sound-field. (a) At each time-frame, we split the signals recorded by each microphone into the same set of frequency subbands. (b) Based on time-difference of arrival estimation between all pairs of recordings, we sample all corresponding hyperbolic loci to obtain a position estimate for the considered subband. (c) Position estimates for all subbands at the considered time-frame (shown as colored spheres).

beamforming strategy [19, 14, 51]. In our case, the use of freely positioned microphones, which may be widely spaced, prevents from using a beamforming strategy. Besides, such an approach would only lead to direction of arrival information and not a 3D position (unless several beamforming arrays were used simultaneously). In our context, we chose to use a TDOA strategy to determine the location of the various auditory events. Since we do not know the directivity of the sound sources nor the response of the microphones, localization based on level difference cannot be applied.

Figure 4 details the various stage of our source localization pipeline.

6.1 Time-frequency correlation analysis

Analysis of the recordings is done on a frame by frame basis using short time-windows (typically 20ms long or 1024 samples at CD quality). For a given source position and a given pair of microphones, the propagation delay from the source to the microphones generates a measurable time-difference of arrival. The set of points which generate the same TDOA defines an hyperboloid surface in 3D (or an hyperbola in 2D) which foci are the locations of the two microphones (Figure 5 (b)).

In our case, we estimate the TDOAs, $\hat{\tau}_{mn}$, between pairs of microphones $< m,n >$ in each frequency subband k using standard generalized cross-correlation (GCC) techniques in frequency domain [34, 56, 15]:

$$\hat{\tau}_{mn} = \arg\max_{\tau} GCC_{mn}(\tau), \tag{7}$$

where the GCC function is defined as:

$$GCC_{nm}(\tau) = \sum_{z=1}^{Z} \psi_{nm}(z) E\left\{Y_{kn}(z) Y_{km}^{*}(z)\right\} e^{j(2\pi\tau z/Z)}. \tag{8}$$

Y_{kn} and Y_{km} are the $2Z$-point Fourier transforms of the subband signals (see Eq. 2), $E\left\{Y_{kn}(z)Y_{km}^{*}(z)\right\}$ is the cross spectrum and $.^{*}$ denotes the complex conjugate operator.

For the weighting function, ψ, we use the PHAT weighting

which was shown to give better results in reverberant environments [15]:

$$\psi_{mn}(z) = \frac{1}{|Y_n(z)Y_m^*(z)|} \tag{9}$$

Note that phase differences computed directly on the Fourier transforms, e.g. as used in the DUET technique [32, 74], cannot be applied in our framework since our microphones are widely spaced.

We also experimented with an alternative approach based on the average magnitude difference function (AMDF) [46, 13]. The TDOAs are then given as:

$$\hat{\tau}_{nm} = \arg\min_{\tau} AMDF_{nm}(\tau), \tag{10}$$

where the AMDF function is defined as:

$$AMDF_{nm}(\tau) = \frac{1}{Z}\sum_{z=1}^{Z} |y_{kn}(\tau) - y_{km}(k+\tau)| \tag{11}$$

We compute the cross-correlation using vectors of 8192 samples (185 ms at 44.1KHz). For each time-frame, we search the highest correlation peaks (or lowest AMDF values) between pairs of recordings in the time-window defined by the spacing between the corresponding couple of microphones. The corresponding time-delay is then chosen as the TDOA between the two microphones for the considered time-frame.

In terms of efficiency, the complexity of AMDF-based TDOA estimation (roughly $O(n^2)$ in the number n of time-domain samples) makes it unpractical for large time-delays. In our test-cases, running on a *Pentium4 Xeon* 3.2GHz processor, AMDF-based TDOA estimations required about 47 s. per subband for one second of input audio data (using 8 recordings, i.e., 28 possible pairs of microphones). In comparison, GCC-based TDOA estimations require only 0.83 s. per subband for each second of recording.

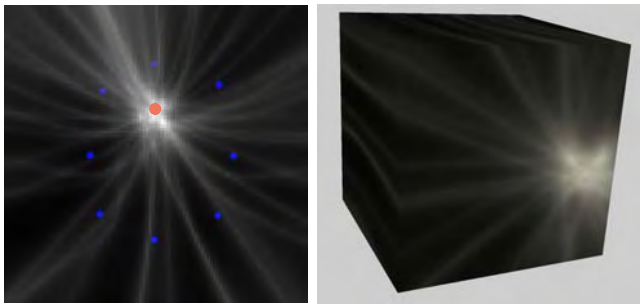
As can be seen in Figure 8, both approaches resulted in comparable subband localization performance and we found both approaches to perform reasonably well in all our test cases. In more reverberant environments, an alternative approach could be the adaptive eigenvalue decomposition [30]. From a perceptual point-of-view, listening to virtual re-renderings, we found that the AMDF-based approach lead to reduced artifacts, which seems to indicate that subband locations are more perceptually valid in this case. However, validation of this aspect would require a more thorough perceptual study.

6.2 Position estimation

From the TDOA estimates, several techniques can be used to estimate the location of the actual sound source. For instance, it can be calculated in a least-square sense by solving a system of equations [30] or by aggregating all estimates into a probability distribution function [60, 1]. Solving for possible positions in a least-square sense lead to large errors in our case, mainly due to the presence of multiple sources, several local maxima for each frequency subband resulting in an averaged localization. Rather, we choose the latter solution and compute a histogram corresponding to the probability distribution function by sampling it on a spatial grid (Figure 6) whose size is defined according to the extent of the auditory environment we want to capture (in our various examples, the grid covered areas ranging from 25 to 400 m^2). We then pick the maximum value in the histogram to obtain the position of the subband.

For each cell in the grid, we sum a weighted contribution of the distance function $D_{ij}(\mathbf{x})$ to the hyperboloid defined by the TDOA for each pair of microphones $< i, j >$:

$$D_{ij}(\mathbf{x}) = |(||M_i - \mathbf{x}|| - ||M_j - \mathbf{x}||) - DDOA_{ij}|, \tag{12}$$



(a) (b)

Figure 6: (a) A 2D probability histogram for source location obtained by sampling a weighted sum of hyperbolas corresponding to the time-difference of arrival to all microphone pairs (shown in blue). We pick the maximum value (in red) in the histogram as the location of the frequency band at each frame. (b) A cut through a 3D histogram of the same situation obtained by sampling hyperboloid surfaces on a 3D grid.

where M_i resp. M_j is the position of microphone i resp. j, $\mathbf{x}$ is the center of the cell and $DDOA_{ij} = TDOA_{ij}/c$ is the signed distance-difference obtained from the calculated TDOA (in seconds) and the speed of sound c.

The final histogram value in each cell is then obtained as :

$$H(\mathbf{x}) = \sum_{ij} \left[\frac{e^{(\gamma(1-D_{ij}(\mathbf{x})))}}{e^{\gamma}} (1 - DDOA_{ij}/||M_i - M_j||) \right. \\ \left. \text{if } D_{ij}(\mathbf{x}) < 1,\ 0 \text{ otherwise} \right]. \tag{13}$$

The exponentially decreasing function controls the "width" of the hyperboloid and provides a tradeoff between localization accuracy and robustness to noise in the TDOA estimates. In our examples, we use $\gamma = 4$. The second weighting term reduces the contribution of large TDOAs relative to the spacing between the pair of microphones. Such large TDOAs lead to "flat" ellipsoids contributing to a large number of neighboring cells in the histogram and resulting into less accurate position estimates [4].

The histogram is re-computed for each subband at each time-frame based on the corresponding TDOA estimates. The location of the k_{th} subband is finally chosen as the center point of the cell having the maximum value in the probability histogram (Figure 5 (c)):

$$B_k = \arg\max_{\mathbf{x}} H(\mathbf{x}) \tag{14}$$

In the case where most of the sound sources and microphones are located at similar height in a near planar configuration, the histogram can be computed on a 2D grid. This yields faster results at the expense of some error in localization. A naive calculation of the histogram at each time-frame (for a single frequency band and 8 microphones, i.e., 28 possible hyperboloids) on a 128×128 grid requires 20 milliseconds on a *Pentium4 Xeon* 3.2GHz processor. An identical calculation in 3D requires 2.3 s. on a $128 \times 128 \times 128$ grid. To avoid this extra computation time, we implemented a hierarchical evaluation using a quadtree or octree decomposition [61]. We recursively test only a few candidate locations (typically 16 to 64), uniformly distributed in each cell, before subdividing the cell in which the maximum of all estimates is found. Our hierarchical localization process supports real-time performance requiring only 5 ms to locate a subband in a $512 \times 512 \times 512$ 3D grid. In terms of accuracy, it was found to be comparable to the direct, non-hierarchical, evaluation at maximum resolution in our test examples.

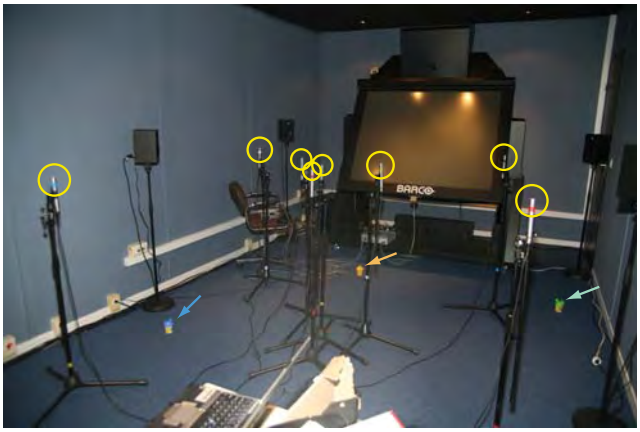


Figure 7: Indoor validation setup using 8 microphones. The 3 markers (see blue, yellow, green arrows) on the ground correspond to the location of the recorded speech signals.

6.3 Indoor validation study

To validate our approach, we conducted a test-study using 8 microphones inside a 7m×3.5m×2.5m room with limited reverberation time (about 0.3 sec. at 1KHz). We recorded three people speaking while standing at locations specified by colored markers. Figure 7 depicts the corresponding setup. We first evaluated the localization accuracy for all subbands by constructing spatial energy maps of the recordings. As can be seen in Figure 8, our approach properly localizes the corresponding sources. In this case, the energy corresponds to the signal captured by a microphone located at the center of the room.

Figure 11 shows localization error over all subbands by reference to the three possible positions for the sources. Since we do not know *a priori* which subband belongs to which source, the error is simply computed, for each subband, as the minimum distance between the reconstructed location of the subband and each possible source position. Our approach achieves a maximum accuracy of one centimeter and, on average, the localization accuracy is of the order of 10 centimeters. Maximum errors are of the order of a few meters. However, listening tests exhibit no strong artefacts showing that such errors are likely to occur for frequency subbands containing very little energy. Figure 11 also shows the energy of one of the captured signals. As can be expected, the overall localization error is also correlated with the energy of the signal.

We also performed informal comparisons between reference binaural recordings and a spatial audio rendering using the obtained locations, as described in the next section. Corresponding audio files can be found at:
`http://www-sop.inria.fr/reves/projects/audioMatting`.

They exhibit good correspondence between the original situation and our renderings showing that we properly assign the subbands to the correct source locations at each time-frame.

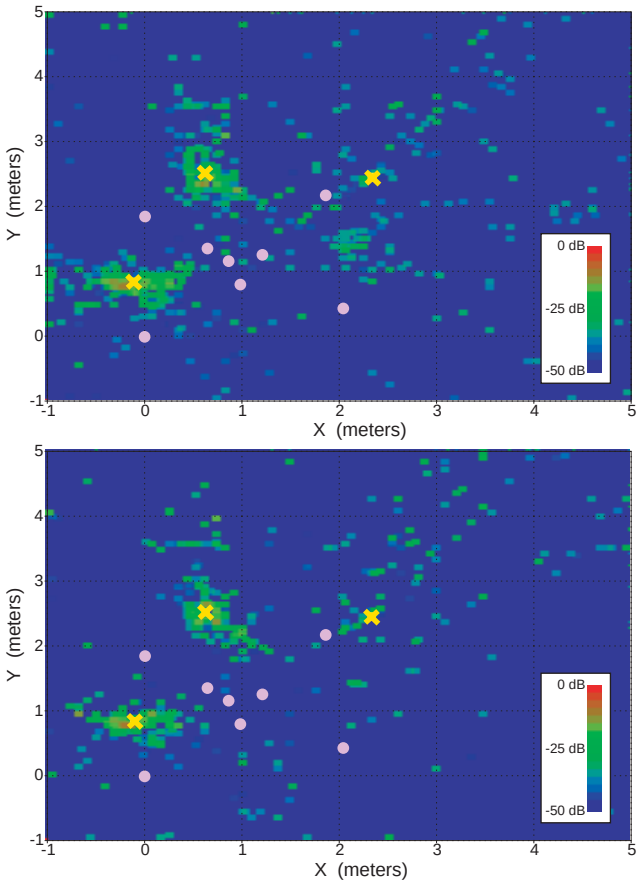


Figure 8: Energy localization map for a 28s.-long audio sequence featuring 3 speakers inside a room (indicated by the three yellow crosses). Light-purple dots show the location of the 8 microphones. The top map is computed using AMDF-based TDOA estimation while the bottom map is computed using GCC-PHAT. Both maps were computed using 8 subbands and corresponding energy is integrated over the entire duration of the sequence.

7 3D-audio resynthesis

The final stage of our approach is the spatial audio resynthesis. During a real-time simulation, the previously pre-computed subband positions can be used for re-rendering the acquired soundfield while changing the position of the sources and listener. A key aspect of our approach is to provide a spatial description of a real-world auditory scene in a manner independent of the auditory restitution system. The scene can thus be re-rendered by standard 3D-audio APIs: in some of our test examples, we used *DirectSound 3D* accelerated by a *CreativeLabs Audigy2 NX* soundcard and also implemented our own software binaural renderer, using head-related

transfer function (HRTF) data from the LISTEN HRTF database[1].

Inspired by binaural-cue coding [23], our re-rendering algorithm can be decomposed in two steps, that we detail in the following sections:

- First, as the virtual listener moves throughout the environment, we construct a *warped monophonic signal* based on the original recording of the microphone closest to the current listening position.

- Second, this warped signal is spatially enhanced using 3D-audio processing based on the location of the different frequency subbands.

These two steps are carried out over small time-frames (of the same size as in the analysis stage). To avoid artefacts we use a 10% overlap to cross-fade successive synthesis frames.

7.1 Warping the original recordings

For re-rendering, a monophonic signal best matching the current location of the virtual listener relative to the various sources must be synthesized from the original recordings.

At each time-frame, we first locate the microphone closest to the location of the virtual listener. To ensure that we remain as faithful as possible to the original recording, we use the signal captured by this microphone as our reference signal $R(t)$.

We then split this signal into the same frequency subbands used during the off-line analysis stage. Each subband is then warped to the virtual listener location according to the pre-computed spatial mapping at the considered synthesis time-frame (Figure 9).

This warping involves correcting the propagation delay and attenuation of the reference signal for the new listening position, according to our propagation model (see Eq.1). Assuming an inverse distance attenuation for point emitters, the warped signal $R'_i(t)$ in subband i is thus given as:

$$R'_i(t) = r^i_1 / r^i_2 R_i(t + (\delta^i_1 - \delta^i_2)), \tag{15}$$

where r^i_1,δ^i_1 are respectively the distance and propagation delay from the considered time-frequency atom to the reference microphone and r^i_2,δ^i_2 are the distance and propagation delay to the new listening position.

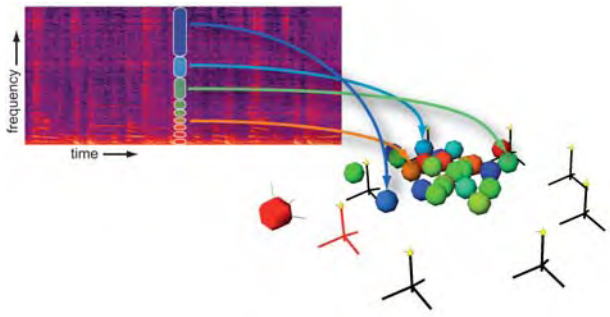


Figure 9: In the resynthesis phase, the frequency components of the signal captured by the microphone closest to the location of the virtual listener (shown in red) is warped according to the spatial mapping pre-computed in the off-line stage.

[1] http://recherche.ircam.fr/equipes/salles/listen/

7.2 Clustering for 3D-audio rendering and source matting

To spatially enhance the previously obtained warped signals, we run an additional clustering step to aggregate subbands which might be located at nearby positions using the technique of [69]. The clustering allows to build groups of subbands which can be rendered from a single representative location and might actually belong to the same physical source in the original recordings. Thus, our final rendering stage spatializes N representative point sources corresponding to the N generated clusters, which can vary between 1 and the total number of subbands. To improve the temporal coherence of the approach we use an additional Kalman filtering step on the resulting cluster locations [33].

With each cluster we associate a weighted sum of all warped signals in each subband which depends on the Euclidean distance between the location of the subband B_i and the location of the cluster representative C_k. This defines matting coefficients α_k, similar to alpha-channels in graphics [54]:

$$\alpha(C_k, B_i) = \frac{1.0/(\varepsilon + ||C_k - B_i||)}{\sum_i \alpha(C_k, B_i)}. \tag{16}$$

In our examples, we used $\varepsilon = 0.1$. Note that in order to preserve the energy distribution, these coefficients are normalized in each frequency subband.

These matting coefficients control the blending of all subbands rendered at each cluster location and help smooth the effects of localization errors. They also ensure a smoother reconstruction when sources are modified or moved around in the re-rendering phase.

The signal for each cluster $S_k(t)$ is finally constructed as a sum of all warped subband signals $R'_i(t)$, as described in the previous section, weighted by the matting coefficients $\alpha(C_k, B_i)$:

$$S_k(t) = \sum_i \alpha(C_k, B_i) R'_i(t). \tag{17}$$

The representative location of each cluster is used to apply the desired 3D-audio processing (e.g., HRTFs) without *a priori* knowledge of the restitution setup.

Figure 10 summarizes the complete re-rendering algorithm.

8 Applications and results

Our technique opens many interesting application areas for interactive 3D applications, such as games or virtual/augmented reality, and off-line audio-visual post-production. Several example renderings demonstrating our approach can be found at the following URL:
`http://www-sop.inria.fr/reves/projects/audioMatting`.

8.1 Modeling complex sound sources

Our approach can be used to render extended sound sources (or small soundscapes) which might be difficult to model using individual point sources because of their complex acoustic behavior. For instance, we recorded a real-world sound scene involving a car which is an extended vibrating sound radiator. Depending on the point of view around the scene, the sound changes significantly due to the relative position of the various mechanical elements (engine, exhaust, etc.) and the effects of sound propagation around the body of the car. This makes an approach using multiple recordings very interesting in order to realistically capture these effects. Unlike other techniques, such as *Ambisonics O-format* [43], our approach captures the position of the various sounding components and not only their directional aspect. In the accompanying examples, we demonstrate a re-rendering with a moving listening point of a car

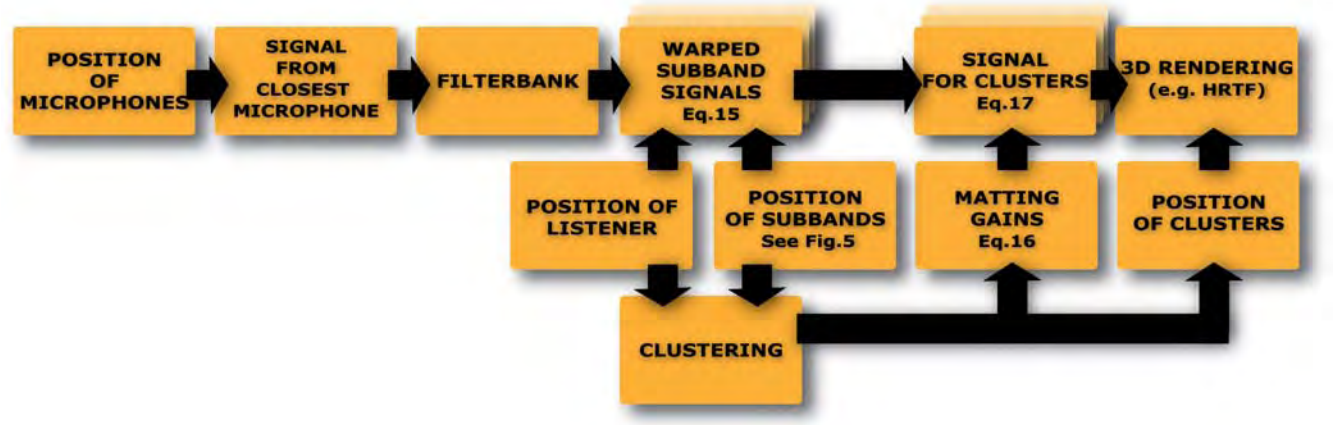
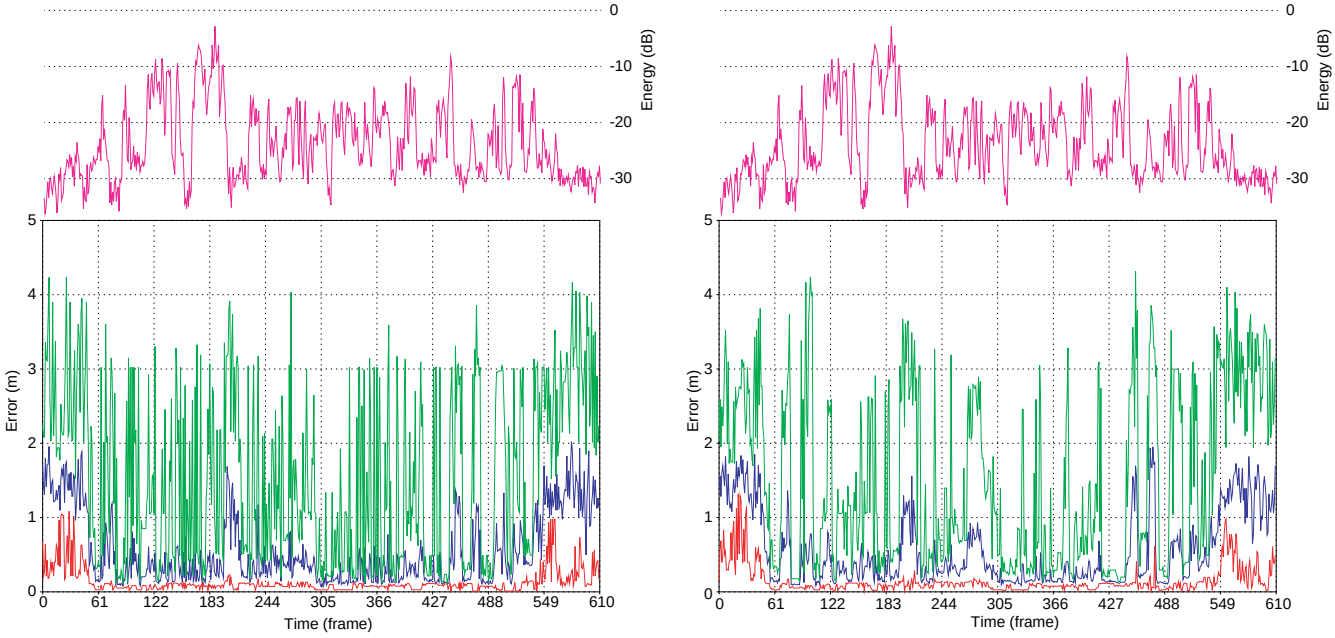


Figure 10: Overview of the synthesis algorithm used to re-render the acquired soundscape based on the previously obtained subband positions.



(a) GCC-PHAT based localization

(b) AMDF-based localization

Figure 11: Localization error for the same audio sequence as in Figure 8. computed over 8 subbands. Averaged error over all subbands is displayed in blue, maximum error in green and minimum error in red. The top (magenta) curve represents the energy for one of the input recordings and shows its correlation with the localization error (clearly larger when the energy drops out).

scenario acquired using 8 microphones surrounding the action (Figure 12). In this case, we used 4 clusters for re-rendering. Note in the accompanying video available on-line, the realistic distance and propagation effects captured by the recordings, for instance on the door slams. Figure 13 shows a corresponding energy map clearly showing the low frequency exhaust noise localized at the rear of the car and the music from the on-board stereo audible through the driver's open window. Engine noise was localized more diffusely mainly due to interference with the music.

Figure 12: We capture an auditory environment featuring a complex sound source (car engine/exhaust, passengers talking, door slams and on-board stereo system) using 8 microphones surrounding the action.

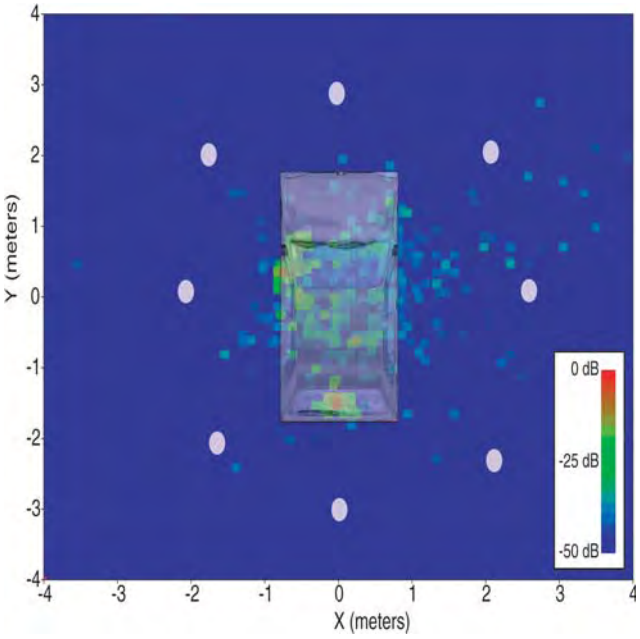


Figure 13: Energy localization map for a 15 sec.-long recording of our car scenario featuring engine/exhaust sounds and music (on the on-board stereo system and audible through the open driver-window). Positions were computed over 8 subbands using GCC-PHAT-based TDOA estimation. Energy is integrated over the entire duration of the input audio sequence.

8.2 Spatial recording and view-interpolation

Following binaural cue coding principles, our approach can be used to efficiently generate high-resolution surround recordings from monophonic signals. To illustrate this application we used 8 omni-directional microphones located in a circle-like configuration about 1.2 meters in diameter (Figure 14) to record three persons talking and the surrounding ambiance (fountain, birds, etc.). Then, our pre-processing was applied to extract the location of the sources. For re-rendering, the monophonic signal of a single microphone was used and respatialized as described in Section 7.1, using 4 clusters (Figure 16). Please, refer to the accompanying video provided on the web site to evaluate the result.

Figure 14: Microphone setup used to record the fountain example. In this case the microphones are placed at the center of the action.

Another advantage of our approach is to allow for re-rendering an acquired auditory environment from various listening points. To demonstrate this approach on a larger environment, we recorded two moving speakers in a wide area (about 15×5 meters) using the microphone configuration shown in Figure 1 (Left). The recording also features several background sounds such as traffic and road-work noises. Figure 15 shows a corresponding spatial energy map. The two intersecting trajectories of the moving speakers are clearly visible.

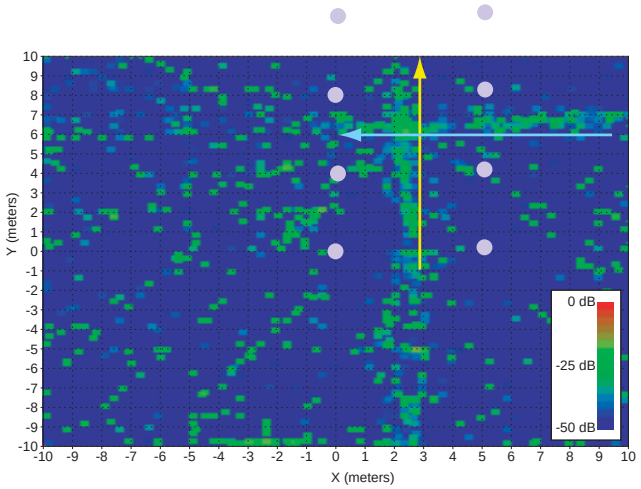


Figure 15: Energy map for a recording of our moving speaker scenario. The arrows depict the trajectory of the two speakers. Energy is integrated over the entire duration of the input audio sequence. Note how the two intersecting trajectories are clearly reconstructed.

Applying our approach, we are able to re-render this auditory scene from any arbitrary viewpoint. Although the rendering is based only on the *monophonic* signal of the microphone closest to the virtual listener at each time-frame, the extracted spatial mapping allows for convincingly reproducing the motion of the sources. Note in the example video provided on the accompanying web-site

how we properly capture front-to-back and left-to-right motion for the two moving speakers.

8.3 Spatial audio compositing and post-editing

Finally, our approach allows for post-editing the acquired auditory environments and composite several recordings.

Source re-localization and modification

Using our technique, we can selectively choose and modify various elements of the original recordings. For instance, we can select any spatial area in the scene and simply relocate all clusters included in the selected region. We demonstrate an example interactive interface for spatial modification where the user first defines a selection area then a destination location. All clusters entering the selection area are translated to the destination location using the translation vector defined by the center of the selection box and the target location. In the accompanying video, we show two instances of source re-localization where we first select a speaker on the left-hand side of the listener and move him to the right-hand side. In a second example, we select the fountain at the rear-left of the listener and move it to the front-right (Figure 16).

Compositing

Since our recording setups are spatially calibrated, we can integrate several environments into a single composite rendering which preserves the relative size and positioning of the various sound sources. For instance, it can be used to integrate a close-miked sound situation into a different background ambiance. We demonstrate an example of sound-field compositing by inserting our previous car example (Figure 12) into the scene with the two moving speakers (Figure 1). The resulting composite environment is rendered with 8 clusters and the 16 recordings of the two original soundscapes. Future work might include merging the representations in order to limit the number of composite recordings (for instance by "re-projecting" the recordings of one environment into the recording setup of the other and mixing the resulting signals).

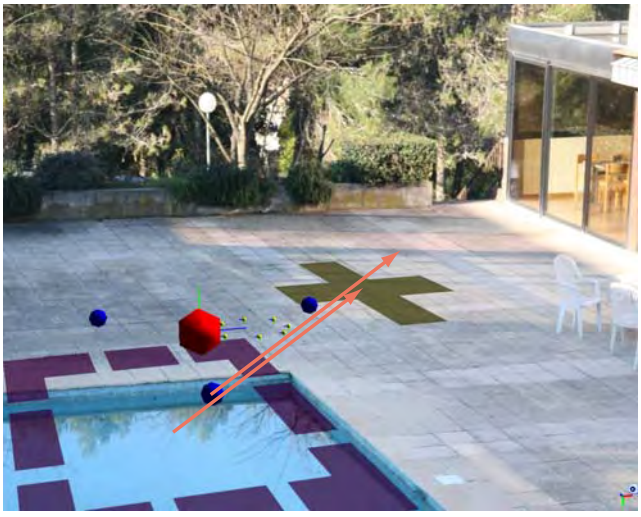


Figure 16: An example interface for source re-localization. In this example we select the area corresponding to the fountain (in purple) and translate it to a new location (shown as a yellow cross). The listener is depicted as a large red sphere, the microphone array as small yellow spheres and the blue spheres show cluster locations.

Real/Virtual integration

Our approach permits spatially consistent compositing of virtual sources within real-world recordings. We can also integrate virtual objects, such as walls, and make them interact with the original recordings. For instance, by performing real-time ray-casting between the listener and the location of the frequency subbands, we can add occlusion effects due to a virtual obstacle using a model similar to [70]. Please, refer to the accompanying examples at the previously mentioned URL for a demonstration. Of course, perfect integration would also require correcting for the reverberation effects between the different environments to composite. Currently, we experimented only in environments with limited reverberation but blind extraction of reverberation parameters [7] and blind deconvolution are complementary areas of future research in order to better composite real and virtual sound-fields.

9 Discussion

Although it is based on a simple mixing model and assumes W-disjoint orthogonality for the sources, we were able to apply our approach to real-world recording scenarios. While not production-grade yet, our results seem promising for a number of interactive and off-line applications.

While we tested it for both indoor and outdoor recordings, our approach is currently only applicable to environments with limited reverberation. Long reverberations will have a strong impact on our localization process since existing cross-correlation approaches are not very robust to interfering sound reflections. Other solutions based on blind channel identification in a reverberant context could lead to improved results [15].

Errors in localization of the frequency subbands can result in noticeable artefacts especially when moving very close to a source. These errors can come from several factors in our examples particularly low signal-to-noise ratio for the source to localize, blurring from sound reflections, correlation of two different signals in the case of widely spaced microphones or several sources being present in a single frequency subband. As a result, several overlapping sources are often fused at the location of the louder source. While the assumption of W-disjoint orthogonality has been proven to be suitable for speech signals [59], it is more questionnable for more general scenarios. It will only be acceptable if this source can perceptually mask the others. However, recent approaches for efficient audio rendering have shown that masking between sources is significant [69], which might explain why our approach can give satisfying results quite beyond the validity domain of the underlying models. Alternate decompositions [45, 40] could also lead to sparser representations and better results within the same framework.

The signal-to-noise ratio of the different sound sources is also directly linked to the quality of the result when moving very close to the source since our warping is likely to amplify the signal of the original recording in this case.

We are working on several improvements to alleviate remaining limitations of the system and improve the rendering quality:

Currently, we do not interpolate between recordings but select the signal of the microphone closest to the listener location for subsequent warping and re-rendering. This provides a correct solution for the case of omnidirectional anechoic point sources. In more general situations, discontinuities might still appear when switching from one microphone to the next. This can be caused, for instance, by the presence of a sound source with a strong directionality. A solution to this problem would be to warp the few microphones closest to the listener and blend the result at the expense of a higher computing cost. Note that naive blending between microphone signals before warping would introduce unwanted interferences, very

noticeable in the case of widely-spaced microphones. Another option would be to experiment with morphing techniques [65] as an alternative to our position-based warping. We could also use different microphones for each frequency subband, for instance choosing the microphone closer to the location of each subband rather than the one closest to the listener. This would increase the signal-to-noise ratio for each source and could be useful to approximate a close-miking situation in order to edit or modify the reverberation effects for instance.

The number of bands also influences the quality of the result. More bands are likely to increase the spatial separation but since our correlation estimates are significantly noisy, it might also make artefacts more audible. In our case, we obtained better sounding results using a limited number of subbands (typically 8 to 16). Following the work of Faller et al. [8, 23, 22], we could also keep track of the inter-correlation between recordings in order to precisely localize only the frames with high correlation. Frames with low correlation could be rendered as "diffuse", forming a background ambiance which cannot be as precisely located [47]. This could be seen as explicitly taking background noise or spatially extended sound sources into account in our mixing model instead of considering only perfect anechoic point sources. We started to experiment with an explicit separation of background noise using noise-removal techniques [21]. The obtained foreground component can then be processed using our approach while the background-noise component can be rendered separately at a lower spatial resolution. Example renderings available on the web site demonstrate improved quality in complex situations such as a seashore recording.

Sound source clustering and matting also strongly depends on the correlation and position estimates for the subbands. An alternative solution would be to first separate a number of sources using independent component analysis (ICA) techniques and then run TDOA estimation on the resulting signals [62, 29]. However, while ICA might improve separation of some sources, it might still lead to signals where sources originating from different locations are combined.

Another issue is the microphone setup used for the recordings. Any number of microphones can be used for localization starting from two (which would only give directional information). If more microphones are used, the additional TDOA estimates will increase the robustness of the localization process. From our experience, closely spaced microphones will essentially return directional information while microphone setups surrounding the scene will give good localization accuracy. Microphones uniformly spaced in the scene provide a good compromise between signal-to-noise ratio and sampling of the spatial variations of the sound-field. We also experimented with cardioid microphone recordings and obtained good results in our car example. However, for larger environments, correlation estimates are likely to become noisier due to the increase in separation between different recordings, making them difficult to correlate. Moreover, it would make interpolating between recordings more difficult in the general case. Our preferred solution was thus to use a set of identical omnidirectional microphones. However, it should be possible to use different sets of microphones for localization and re-rendering which opens other interesting possibilities for content creation, for instance by generating consistent 3D-audio flythroughs while changing the focus point on the scene using directional microphones.

Finally, our approach currently requires an off-line step which prevents it from being used for real-time analysis. Being able to compute cross-correlations in real-time for all pairs of microphones and all subbands would make the approach usable for broadcast applications.

10 Conclusions

We presented an approach to record, edit and re-render real-world auditory situations. Contrary to most related approaches, we acquire the sound-field using an unconstrained, widely-spaced, microphone array which we spatially calibrate using photographs. Our approach pre-computes a spatial mapping between different frequency subbands of the acquired live recordings and the location in space from which they were emitted. We evaluated standard TDOA-based techniques and proposed a novel hierarchical localization approach. At run-time, we can apply this mapping to the frequency subbands of the microphone closest to the virtual listener in order to resynthesize a consistent 3D sound-field, including complex propagation effects which would be difficult to simulate. An additional clustering step allows for aggregating subbands originating from nearby location in order to segment individual sound sources or small groups of sound sources which can then be edited or moved around. To our knowledge, such level of editing was impossible to achieve using previous state-of-the-art and could lead to novel authoring tools for 3D-audio scenes.

We believe our approach opens many novel perspectives for interactive spatial audio rendering or off-line post-production environments, for example to complement image based rendering techniques or free-viewpoint video. Moreover, it provides a compact encoding of the spatial sound-field, which is independent of the restitution system. In the near future, we plan to run more formal perceptual tests in order to compare our results to binaural or high-order *Ambisonics* recordings in the case of fixed-viewpoint scenarios and to evaluate its quality using various restitution systems. From a psychophysical point of view, this work suggests that real-world sound scenes can be efficiently encoded using limited spatial information.

Other promising areas of future work would be to exploit perceptual localization results to improve localization estimation [72] and apply our analysis-synthesis strategy to the real-time generation of spatialized audio textures [42]. Finaly, making the calibration and analysis step interactive would allow the approach to be used in broadcasting applications (e.g., 3D TV).

Acknowledgments

This research was made possible by a grant from the *région PACA* and was also partially funded by the RNTL project OPERA (http://www-sop.inria.fr/reves/OPERA). We acknowledge the generous donation of *Maya* as part of the *Alias* research donation program, Alexander Olivier-Mangon for the initial model of the car, and Frank Firsching for the animation.

References

[1] P. Aarabi. The fusion of distributed microphone arrays for sound localization. *EURASIP Journal on Applied Signal Processing*, 4:338–347, 2003.

[2] T.D. Abhayapala and D.B. Ward. Theory and design of high order sound field microphones using spherical microphone array. *Proceedings of Intl. Conf. on Acoustics, Speech and Signal Processing*, 2002.

[3] T. Ajdler and M. Vetterli. The plenacoustic function and its sampling. *Proc. of the 1st Benelux Workshop on Model-based processing and coding of audio (MPCA2002), Leuven, Belgium*, November 2002.

[4] Thibaut Ajdler, Igor Kozintsev, Rainer Lienhart, and Martin Vetterli. Acoustic source localization in distributed sensor networks. *Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA*, 2:1328–1332, 2004.

[5] Daniel G. Aliaga and Ingrid Carlbom. Plenoptic stitching: a scalable method for reconstructing 3d interactive walk throughs. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 443–450, New York, NY, USA, 2001. ACM Press.

[6] C. Avendano. Frequency-domain source identification and manipulation in stereo mixes for enhancement, suppresssion and re-panning applications. *Proc. of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA2003), New Paltz, NY, USA*, October 2003.

[7] A. Baskind and O. Warusfel. Methods for blind computational estimation of perceptual attributes of room acoustics. *proceedings of the AES 22nd Intl. Conf. on virtual, synthetic and entertainment audio, Espoo, Finland*, June 2002.

[8] Frank Baumgarte and Christof Faller. Binaural cue coding - part I: Psychoacoustic fundamentals and design principles. *IEEE Trans. on Speech and Audio Proc*, 11(6), 2003.

[9] A.J. Berkhout, D. de Vries, and P. Vogel. Acoustic control by wave field synthesis. *J. of the Acoustical Society of America*, 93(5):2764–2778, may 1993.

[10] M.M. Boone, E.N.G. Verheijen, and P.F. van Tol. Spatial sound-field reproduction by wave-field synthesis. *J. of the Audio Engineering Society*, 43:1003–1011, December 1995.

[11] A.S. Bregman. *Auditory Scene Analysis, The perceptual organization of sound*. The MIT Press, 1990.

[12] Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. Unstructured lumigraph rendering. *Proc. of ACM SIGGRAPH*, 2001.

[13] J. Chen, J. Benesty, and Y. Huang. Performance of GCC- and AMDF-based time-delay estimation in practical reverberant environments. *EURASIP Journal on Applied Signal Processing*, 1:25–36, 2005.

[14] J.C. Chen, K. Yao, and R.E. Hudson. Acoustic source localization and beamforming: Theory and practice. *EURASIP Journal on Applied Signal Processing*, 4:359–370, 2003.

[15] Jingdong Chen, Jacob Benesty, and Yiteng (Arden) Huang. Time delay estimation in room acoustic environments: An overview. *EURASIP Journal on Applied Signal Processing*, 2006:Article ID 26503, 2006.

[16] S.E. Chen and L. Williams. View interpolation for image synthesis. *Computer Graphics*, 27(Annual Conference Series, Proc. of ACM SIGGRAPH93):279–288, 1993.

[17] P. Comon. Independent component analysis: A new concept. *Signal Processing*, 36:287–314, 1994.

[18] J. Daniel, J.-B. Rault, and J.-D. Polack. Ambisonic encoding of other audio formats for multiple listening conditions. *105th AES convention, preprint 4795*, August 1998.

[19] J.H. DiBiase, H.F. Silverman, and M.S. Branstein. *Microphone Arrays, Signal Processing Techniques and Applications, Chapter 8*. Springer Verlag, 2001.

[20] M.N. Do. Toward sound-based synthesis: the far-field case. *Proc. of IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), Montreal, Canada*, May 2004.

[21] Y. Ephraim and D. Malah. Speech enhancement using a minimum mean-square error short-time spectral amplitude estimator. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 32(6):1109–1121, December 1984.

[22] C. Faller and J. Merimaa. Source localization in complex listening situations: Selection of binaural cues based on interaural coherence. *J. of the Acoustical Society of America*, 116(5):3075–3089, November 2005.

[23] Christof Faller and Frank Baumgarte. Binaural cue coding - part II: Schemes and applications. *IEEE Trans. on Speech and Audio Proc*, 11(6), 2003.

[24] O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. The MIT Press, Cambridge, Mass., 1993.

[25] M.A. Gerzon. Ambisonics in multichannel broadcasting and video. *J. of the Audio Engineering Society*, 33(11):859–871, 1985.

[26] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 43–54, New York, NY, USA, 1996. ACM Press.

[27] U. Horbach, A. Karamustafaoglu, R. Pellegrini, P. Mackensen, and G. Theile. Design and applications of a data-based auralization system for surround sound. *106th Convention of the Audio Engineering Society, preprint 4976*, 1999.

[28] Youichi Horry, Ken-Ichi Anjyo, and Kiyoshi Arai. Tour into the picture: using a spidery mesh interface to make animation from a single image. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 225–232, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[29] G. Huang, L. Yang, and Z. He. Multiple acoustic sources location based on blind source separation. *Proc. of the First International Conference on Natural Computation (ICNC'05)*, 2005.

[30] Y. Huang, J. Benesty, and G.W. Elko. Microphone arrays for video camera steering. *Acoustic Signal Processing for Telecommunications*, 2000.

[31] J.-M. Jot, V. Larcher, and J.-M. Pernaux. A comparative study of 3D audio encoding and rendering techniques. *Proceedings of the AES 16th international conference, Spatial sound reproduction, Rovaniemi, Finland*, april 1999.

[32] Alexander Jourjine, Scott Rickard, and Ozgur Yilmaz. Blind separation of disjoint orthogonal signals: Demixing n sources from 2 mixtures. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'00), Istanbul, Turkey*, June 2000.

[33] R.E. Kalman. A new approach to linear filtering and prediction problems. *Transaction of the ASME-Journal of Basic Engineering*, 82 (Series D):35–45, 1960.

[34] C.H. Knapp and G.C. Carter. The generalized correlation method for estimation of time delay. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 24(4):320–327, August 1976.

[35] H. Krim and M. Viberg. Two decades of array signal processing research. *IEEE Signal Processing Magazine*, pages 67–93, July 1996.

[36] A. Laborie, R. Bruno, and S. Montoya. A new comprehensive approach of surround sound recording. *Proc. 114th convention of the Audio Engineering Society, preprint 5717*, 2003.

[37] A. Laborie, R. Bruno, and S. Montoya. High spatial resolution multi-channel recording. *Proc. 116th convention of the Audio Engineering Society, preprint 6116*, 2004.

[38] Martin J. Leese. Ambisonic surround sound FAQ (version 2.8), 1998. http://members.tripod.com/martin_leese/Ambisonic/.

[39] Marc Levoy and Pat Hanrahan. Light field rendering. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42, New York, NY, USA, 1996. ACM Press.

[40] M. S. Lewicki and T. J. Sejnowski. Learning overcomplete representations. *Neural Computation*, 12(2):337–365, 2000.

[41] M.S. Lewicki. Efficient coding of natural sounds. *Nature Neuroscience*, 5(4):356–363, 2002.

[42] L. Lu, L. Wenyin, and H.-J. Zhang. Audio textures: Theory and applications. *IEEE Transactions on Speech and Audio Processing*, 12(2):156–167, 2004.

[43] D.G. Malham. Spherical harmonic coding of sound objects - the ambisonic 'O' format. *Proc. of the 19th AES Conference, Surround Sound, Techniques, Technology and Perception, Schloss Elmau, Germany*, June 2001.

[44] D.G. Malham and A. Myatt. 3D sound spatialization using ambisonic techniques. *Computer Music Journal*, 19(4):58–70, 1995.

[45] S. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993.

[46] J. Merimaa. Applications of a 3D microphone array. *112th AES convention, preprint 5501*, May 2002.

[47] J. Merimaa and V. Pullki. Spatial impulse response rendering. *Proc. of the 7th Intl. Conf. on Digital Audio Effects (DAFX'04), Naples, Italy*, October 2004.

[48] J. Meyer and G. Elko. Spherical microphone arrays for 3d sound recording. *chap. 2 in Audio Signal Processing for next-generation multimedia communication systems, Eds. Yiteng (Arden) Huang and Jacob Benesty, Bosten, Kluwer Academic Publisher*, 2004.

[49] Brian C.J. Moore. *An introduction to the psychology of hearing*. Academic Press, 4th edition, 1997.

[50] Randolph L. Moses, Dushyanth Krishnamurthy, and Robert Patterson. An auto-calibration method for unattended ground sensors. *Acoustics, Speech, and Signal Processing (ICASSP '02)*, 3:2941– 2944, May 2002.

[51] B. Mungamuru and P. Aarabi. Enhanced sound localization. *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics*, 34(3), June 2004.

[52] P.D. O'Grady, B.A. Pearlmutter, and S.T. Rickard. Survey of sparse and non-sparse methods in source separation. *Intl. Journal on Imaging Systems and Technology (IJIST), special issue on Blind source separation and deconvolution in imaging and image processing*, 2005.

[53] R.S. Pellegrini. Comparison of data and model-based simulation algorithms for auditory virtual environments. *106th Convention of the Audio Engineering Society, preprint 4953*, 1999.

[54] T. Porter and T. Duff. Compositing digital images. *Proceedings of ACM SIGGRAPH 1984*, pages 253–259, July 1984.

[55] V. Pulkki. Directional audio coding in spatial sound reproduction and stereo upmixing. *Proc. of the AES 28th Int. Conf, Pitea, Sweden*, June 2006.

[56] D.V. Rabinkin, R.J. Renomeron, J.C. French, and J.L. Flanagan. Estimation of wavefront arrival delay using the cross-power spectrum phase technique. *132th meeting of the Acoustical Society of America, Honolulu*, December 1996.

[57] Richard Radke and Scott Rickard. Audio interpolation. In *the Audio Engineering Society 22nd International Conference on Virtual, Synthetic and Entertainment Audio (AES'22), Espoo, Finland*, pages 51–57, June 15-17 2002.

[58] S. Rickard. Sparse sources are separated sources. *Proceedings of the 16th Annual European Signal Processing Conference, Florence, Italy*, 2006.

[59] S. Rickard and O. Yilmaz. On the approximate w-disjoint orthogonality of speech. *Proceedings of Intl. Conf. on Acoustics, Speech and Signal Processing*, 2002.

[60] Y. Rui and D. Florencio. New direct approaches to robust sound source localization. *Intl. Conf. on Multimedia and Expo (ICME)*, July 2003.

[61] H. Samet. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley, 1990.

[62] H. Saruwatari, S. Kurita, K. Takeda, F. Itakura, T. Nishikawa, and K. Shikano. Blind source separation combining independent component analysis and beamforming. *EURASIP Journal on Applied Signal Processing*, 11:1135–1146, 2003.

[63] H. Sawada, S. Araki, R. Mukai, and S. Makino. Blind extraction of dominant target sources using ica and time-frequency masking. *IEEE Trans. Audio, Speech, and Language Processing*. accepted for future publication.

[64] R.O. Schmidt. Multiple emitter location and signal parameter estimation. *IEEE Transactions on Antennas and Propagation*, AP-34(3), March 1986.

[65] M. Slaney, M. Covell, and B. Lassiter. Automatic audio morphing. *Proceedings of Intl. Conf. on Acoustics, Speech and Signal Processing*, May 1996.

[66] Soundfield. http://www.soundfield.com.

[67] R. Streicher. The decca tree – it's not just for stereo anymore. http://www.wesdooley.com/pdf/Surround_Sound_Decca_Tree-urtext.pdf.

[68] R. Streicher and F.A. Everest, editors. *The new stereo soundbook, 2nd edition*. Audio Engineering Associate, Pasadena (CA), USA, 1998.

[69] N. Tsingos, E. Gallo, and G. Drettakis. Perceptual audio rendering of complex virtual environments. *ACM Transactions on Graphics, Proceedings of SIGGRAPH 2004*, August 2004.

[70] Nicolas Tsingos and Jean-Dominique Gascuel. Fast rendering of sound occlusion and diffraction effects for virtual acoustic environments. *Proc. 104th Audio Engineering Society Convention, preprint 4699*, May 1998.

[71] E. Vincent, X. Rodet, A. Röbel, C. Févotte, E. Le Carpentier, R. Gribonval, L. Benaroya, and F. Bimbot. A tentative typologie of audio source separation tasks. *Proc. of the 4th Intl. Symposium on Independent Component Analysis and Blind Signal Separation (ICA2003), Nara, Japan*, April 2003.

[72] K.W. Wilson and T. Darell. Learning a precedence effect-like weighting function for the generalized cross-correlation framework. *IEEE Journal of speech and audio processing. Special issue on statistical and perceptual audio processing*, 2006.

[73] D.L. Yewdall. *Practical Art of Motion Picture Sound (2nd edition)*. Focal Press, 2003.

[74] Ozgur Yilmaz and Scott Rickard. Blind separation of speech mixtures via time-frequency masking. *IEEE Transactions on Signal Processing*, 52(7):1830–1847, 2004.

Extracting and Re-rendering Structured Auditory Scenes from Field Recordings

Emmanuel Gallo[1,2] and Nicolas Tsingos[1]

[1]*REVES, INRIA, Sophia Antipolis, France*

[2]*CSTB, Sophia Antipolis, France*

Correspondence should be addressed to Emmanuel Gallo — Nicolas Tsingos
(emmanuel.gallo|nicolas.tsingos@sophia.inria.fr)

ABSTRACT

We present an approach to automatically extract and re-render a structured auditory scene from field recordings obtained with a small set of microphones, freely positioned in the environment. From the recordings and the calibrated position of the microphones, the 3D location of various auditory events can be estimated together with their corresponding content. This structured description is reproduction-setup independent. We propose solutions to classify foreground, well-localized sounds and more diffuse background ambiance and adapt our rendering strategy accordingly. Warping the original recordings during playback allows for simulating smooth changes in the listening point or position of sources. Comparisons to reference binaural and B-format recordings show that our approach achieves good spatial rendering while remaining independent of the reproduction setup and offering extended authoring capabilities.

1. INTRODUCTION

Current models for interactive 3D audio scene authoring often assume that sounds are emitted by a set of monophonic point sources for which a signal has to be individually generated [33, 5]. In the general case, source signals cannot be completely synthesized from physics principles and must be individually recorded, which requires enormous time and resources. Although this approach gives the user the freedom to control each source and freely navigate throughout the auditory scene, the overall result remains an approximation. This is due to the complexity of real-world sources, limitations of microphone pick-up patterns and limitations of the simulated sound propagation models. On the opposite end of the spectrum, spatial sound recording techniques which encode directional components of the soundfield [34, 35, 23, 25] can be directly used to acquire and playback real-world auditory environments as a whole. They produce realistic results but offer little control, if any, at the playback end. In particular, they are acquired from a single location in space and only encode directional information, which makes them insufficient for free-walkthrough applications or rendering of large near-field sources. In such spatially-extended cases, correct reproduction requires sampling the soundfield at several locations and encoding the 3D position and not only the incoming direction of the sounds. In practice, the use of such single-point recordings is mostly limited to the rendering of an overall surround ambiance that can possibly be rotated around the listener.

We previously developed a novel analysis-synthesis approach which bridges the two previous strategies [12]. Inspired by spatial audio coding [10, 4, 6, 28, 13] and blind source separation [38, 37, 30], our method builds a higher-level spatial description of the auditory scene from a small set of monophonic recordings. Contrary to previous spatial audio coding work, the recordings are made from widely-spaced locations and sample both content and spatial information for the sound sources present in the scene. Our approach is also mostly dedicated to live recordings since it reconstructs estimates of the 3D locations of the sound sources from physical propagation delays. This information might not be available in studio recordings which rely on non-physical panning strategies. The obtained description can then be

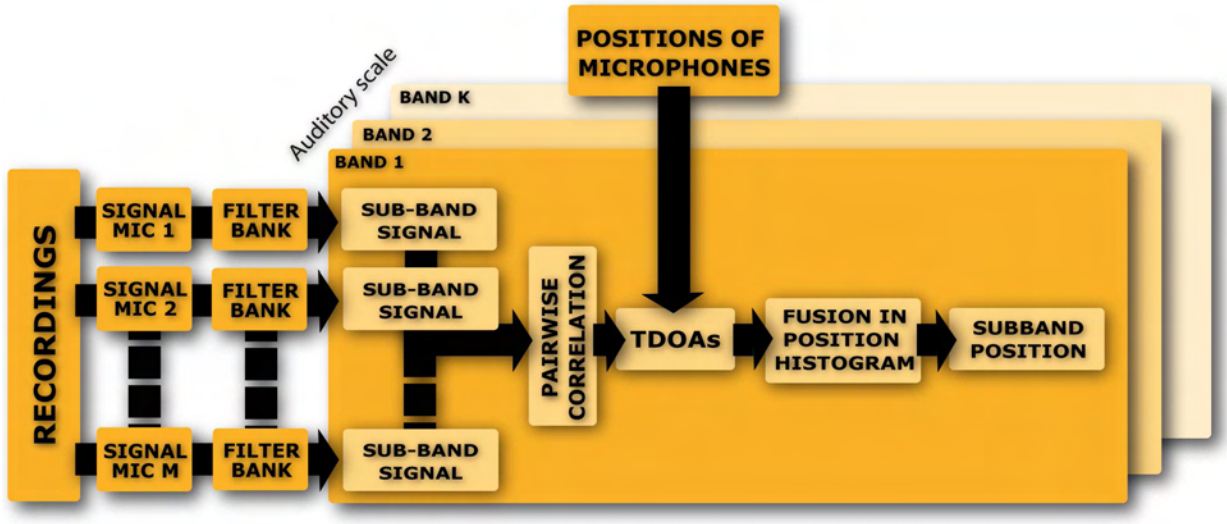


Fig. 1: Overview of our analysis pipeline.

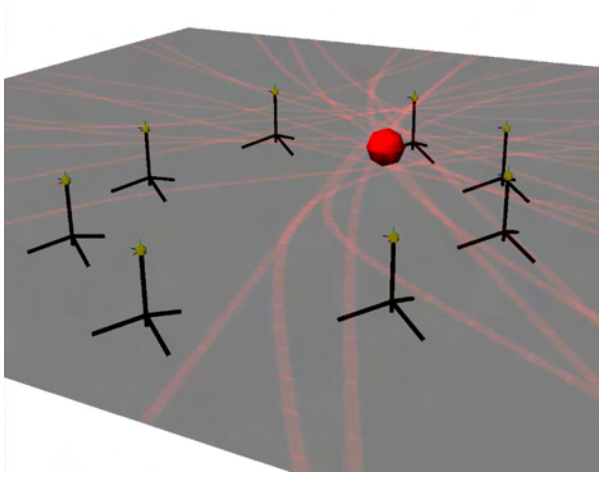


Fig. 2: Construction of a global spatial mapping for the captured sound-field. Based on calculated time-differences of arrival estimation between all pairs of recordings, we sample all corresponding hyperbolic loci to obtain a position estimate for each considered subband (shown as colored sphere).

used for real-time post-processing and re-rendering of the original recordings, for instance by smoothly varying the listening point inside the environment and editing/moving sound sources. We briefly review key aspects of this approach but refer the reader to [12] for additional details.

We first acquire real-world soundscapes using a small number (e.g., 8) of omnidirectional microphones arbitrarily positioned in the environment. In order to extract correct spatial information from the recordings, it is necessary to retrieve the 3D locations of the microphones. We use an image-based technique from photographs [11] which ensures fast and convenient acquisition on location, not requiring any physical measurements or homing device.

The obtained sparse sampling of the soundfield is analyzed in an off-line pre-processing step in order to segment various auditory components and associate them with the position in space from which they were emitted (Figures 1 and 2). To compute this spatial mapping, we split the signal into short time-frames and decompose them onto a set of frequency subbands defined on a Bark scale [36] or, alternatively, using a gammatone filter bank [26]. Assuming that the sound sources do not overlap in time-frequency domain (*W-disjoint orthogonality* hypothesis [38]), we then use classical time-difference of arrival techniques (e.g., [18, 7]) between all pairs of microphones to retrieve a position for each subband at

each time-frame. We developed an improved hierarchical source localization technique from the obtained time-differences, using a quadtree or octree decomposition of space [32].

Real-time re-rendering is achieved through a frequency-dependent warping of the original recordings, based on the estimated positions of each frequency subband. This warping assumes an omnidirectional, anechoic, point source model. For instance, for any desired virtual listener position we first determine the closest microphone and use its signal as a reference. We then warp this reference signal by resampling and equalizing its different subbands. This warping first compensates the original propagation delay and attenuation from each calculated subband location to the location of the reference microphone. It then applies the updated propagation delay and attenuation corresponding to the new position of the virtual listener. Finally, the obtained monophonic signal is enhanced by the spatial cues computed for each subband (e.g., using head-related transfer functions), creating a spatialized rendering. Example re-renderings are available at http://www-sop.inria.fr/reves/projects/audioMatting.

However, in the case of live field recordings, this approach suffers from several limitations. First, the underlying hypothesis of time-frequency sparseness for the acquired signals is often not true in practice, especially in the presence of significant background noise [31]. This results in noisy position estimates and low quality signal reconstruction when virtually moving throughout the environment. Second, our approach uses a limited number of frequency subbands, acting as representative point sources, to model the auditory environment at each time-frame. While point sources might be appropriate to render well-localized events, background ambiance and extended sources (e.g., the sea on a seashore) cannot be convincingly reproduced using this model (Figure 3).

In this paper, we propose a solution to these shortcomings based on an *a priori* segmentation of foreground sound events and background ambiance which we describe in Section 2.1. We also present an improved re-rendering solution specifically adapted to these two components which preserves the independence from the reproduction setup. In particular, we propose to render the foreground sound events using a set of separate point sources while the background component is encoded using a smoother low-order spherical harmonics representation. Details can be found in Sections 2.2 and 2.3.

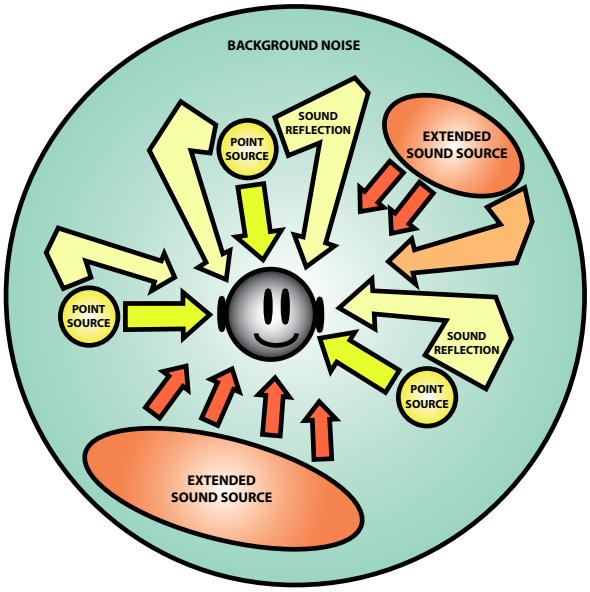


Fig. 3: Typical components of a real-world auditory scene. In this paper, we propose to explicitly separate *foreground*, non-stationary and well localized, sound events from *background* components that are more stationary and spatially diffuse.

Section 3 describes the results of a pilot perceptual evaluation study aimed at assessing the quality of our approach relative to reference binaural and B-format recordings in the case of fixed-listening-point scenarios.

Finally, our approach introduces additional authoring capabilities by allowing separate manipulation of each component, which we briefly outline in Section 4 before concluding.

2. IMPROVED ANALYSIS AND RE-SYNTHESIS

This section addresses a set of possible improvements to our previous technique. They are based on an *a priori* segmentation of background and foreground components leading to a two-layer model, similar in spirit to the pairwise/non-directional and direct/diffuse decompositions used in some spatial audio coding approaches [28, 13, 29, 6]. However, since we are warping the direct component when re-rendering from different listening points, switching between localized/diffuse models on a per-subband basis would introduce audible artefacts in our case. We chose to perform a finer-grain segmentation

of the input recordings as a pre-processing step which does not rely on position estimates. Such an approach was already reported to improve results for blind source separation problems [8]. We also propose re-rendering strategies tailored to each component.

2.1. Background/foreground segmentation

We chose to segment stationary background noise from non-stationary sound events using the technique by Ephraim and Malah [9], originally developed for denoising of speech signals. This approach assumes that the distributions of Fourier coefficients for signal and noise are statistically independent zero-mean Gaussian random variables. Under this assumption, the spectral amplitude of the denoised signal is estimated using a minimum mean-square error criterion. The background noise signal is then simply obtained by subtracting the denoised signal from the original. We found the algorithm to perform quite well. While not perfect, it leads to a foreground component with limited musical noise. In most cases, this noise is masked when re-combined with the background component at re-rendering time. The extracted foreground component, containing non-stationary sounds is also better suited to our underlying assumption of time-frequency sparseness than the original recordings (see Figure 5). However, several foreground sound sources might still overlap in time-frequency. Background and foreground segmentation is performed independently on the signals from all microphones.

2.2. Background "panorama" generation

The separated foreground and background components are both processed using the analysis pipeline described in Section 1 (see also Figure 1). However, in the case of the background component, we obtain noisier position estimates since this component will generally correspond to background noise and sources with low signal-to-noise ratios. In order to produce a smooth spatial background texture, we use the obtained positions to encode the corresponding subband signals on a 1^{st}-order spherical harmonic basis. No warping is applied to the background component in this case (Figure 4).

As our signals are real-valued, we encode them with real spherical harmonics defined as:

$$y_l^m(\theta,\phi) = \begin{cases} \sqrt{2}K_l^m cos(m\phi)P_l^m(cos\theta) & m>0 \\ \sqrt{2}K_l^m cos(-m\phi)P_l^{-m}(cos\theta) & m<0 \\ K_l^0 P_l^0(cos\theta) & m=0 \end{cases} \quad (1)$$

where l is the order, $m \in [-l;+l]$, P is the associated Legendre polynomial and K is a scaling factor defined as:

$$K_l^m = \sqrt{\frac{(2l+1)}{4\pi}\frac{(l-|m|)!}{(l+|m|)!}}. \quad (2)$$

For each subband signals, we compute the minimum and maximum elevation and azimuth of the obtained positions over the entire duration of the recording. Then, we uniformly expand the background signal in this area. We choose the background signal to encode from the monophonic recording closest to the center of the acquired scene. Accordingly, the background texture is encoded relative to a fixed reference point, for instance the central point of the scene.

This background panorama can thus be encoded in a pre-processing stage so that only the decoding is performed at run-time, e.g., when freely navigating in the recordings. Several decoding options are available depending on the desired reproduction setup [15].

2.3. Improved foreground re-synthesis

At re-rendering time, we perform a warping of the original foreground recordings in order to generate a signal as consistent as possible with the desired virtual listening position (Figure 4). Assuming an inverse distance attenuation for point emitters, the warped signal $R'_i(t)$ in subband i is given as:

$$R'_i(t) = r_1^i / r_2^i R_i(t + (\delta_1^i - \delta_2^i)), \quad (3)$$

where r_1^i,δ_1^i are respectively the distance and propagation delay from the considered time-frequency atom to the reference microphone and r_2^i,δ_2^i are the distance and propagation delay to the desired listening position.

This warping heavily relies on the fact that we consider the subband signals to be re-emitted by anechoic point sources. In real-world environments this model is challenged, due to the strong directionality of some sound sources. As a result, discontinuities can appear when the virtual listener is moving around if the signal from a single reference microphone is used (e.g., the one closest to

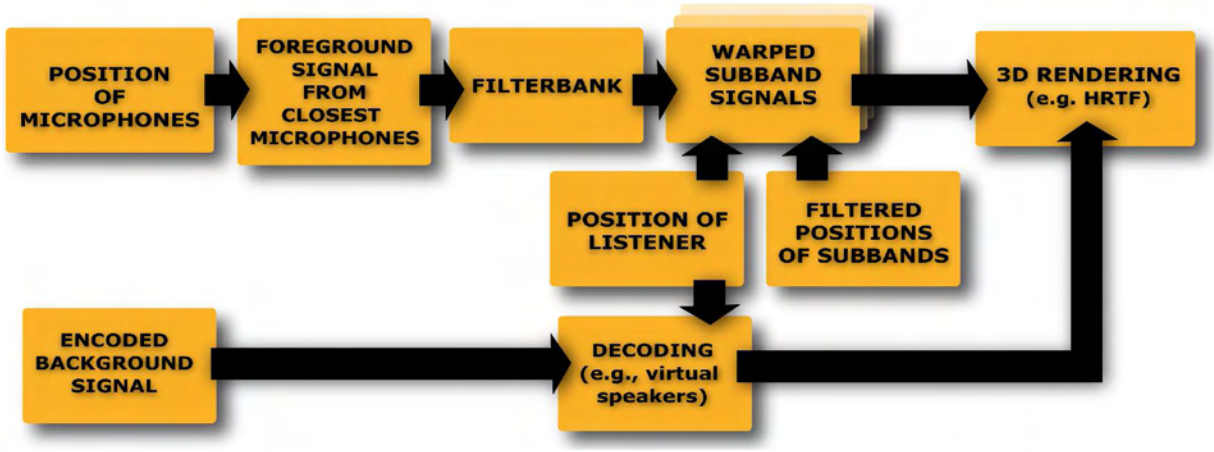


Fig. 4: Overview of our re-synthesis pipeline. Foreground sound events are rendered as point sources while background sounds are encoded using a low-order spherical harmonics decomposition.

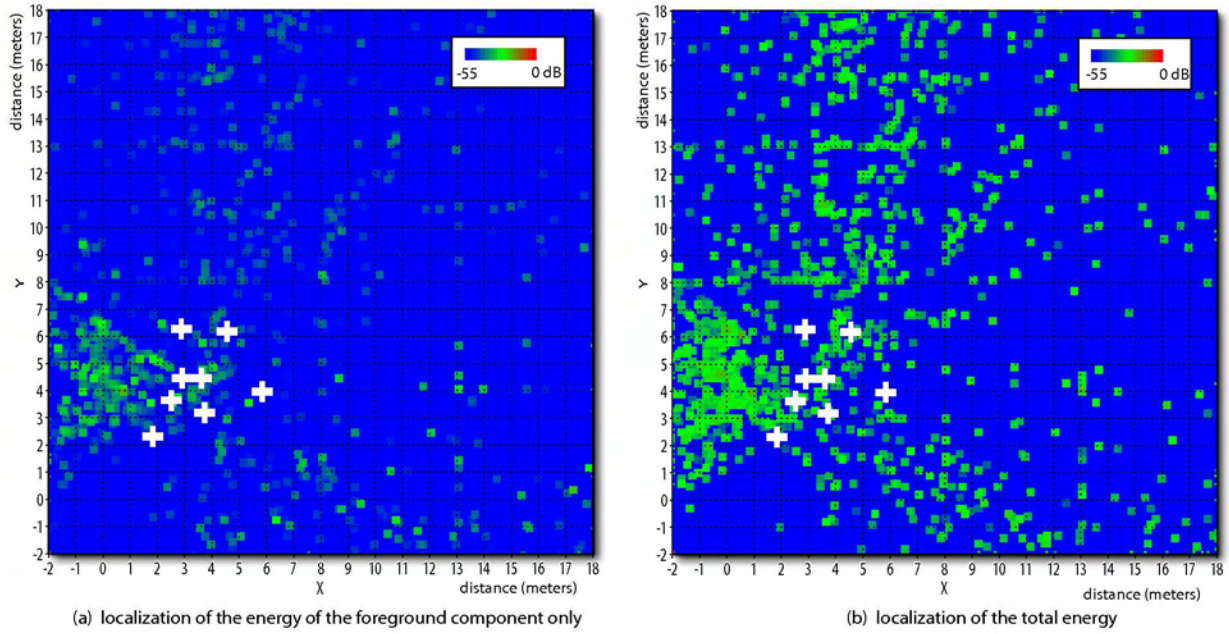


Fig. 5: Comparison between energy localization in the seashore example of Section 4 for (a) the foreground component only and (b) the complete recording. The figure shows the reconstructed location of all subbands integrated through the entire duration of the sequence. White crosses indicate the locations of the microphones used for recording.

the desired virtual position). To avoid such problems and roughly compensate for the limitations of our anechoic point source model, we propose to continuously warp the signals of the two microphones closest to the desired virtual listening position and blend them together to generate a smoothly varying monophonic signal. Blending can be simply controlled by the relative distance of the virtual listener to these two reference microphones. Note that blending the signals prior to warping would introduce comb filtering effects that can be very noticeable

when the microphones are widely spaced. To further improve the re-rendering quality of the foreground component, we also smooth our position estimates for the subbands using Kalman filtering [16]. This prevents large and fast position changes and limits possible "wobbling" effects due to jittery subband positions.

3. PILOT SUBJECTIVE EVALUATION

In order to evaluate the quality of a spatial audio reproduction system based on our approach, we compared it to binaural and B-format recordings in the context of various scenarios with fixed listening points.

3.1. Test stimuli and procedure

We recorded test scenarios in two different environments: indoors in a moderately reverberant room (RT60 $\approx$ 0.3 sec. at 1KHz) and outdoors (see Figure 6). For each scenario, we used 8 monophonic recordings made with *AudioTechnica 3032* omnidirectional microphones to run our localization and re-rendering approach. A pair of *Sennheiser MKE-2 gold* microphones was placed inside the ears of a subject to capture reference binaural recordings and we also acquired a B-format version of the scenes using a *Soundfield ST250* microphone. Eventually, four recordings (one indoors, three outdoors), each about 50 sec. long, were chosen for quality testing.

We used 8 non-overlapping subbands uniformly distributed on a Bark scale to run our spatial analysis (Figure 1). Then, a binaural rendering from a point of view similar to the binaural and B-format recordings was generated from the monophonic input of the closest omnidirectional microphone and the time-varying locations obtained for the subbands. The signal of the same microphone was used to generate both a binaural rendering of the foreground events and the 1^{st}-order spherical harmonic background decoded over headphones using a *virtual loudspeakers* technique. In both cases, we used head related transfer functions (HRTFs) of the *LISTEN* database (http://recherche.ircam.fr/equipes/salles/listen/) for re-rendering. We also generated a re-rendering without explicit background/foreground segmentation considering the original recording to be entirely foreground. B-format recordings were also converted to binaural using a similar virtual loudspeaker approach.

We used a protocol derived from *Multiple Stimuli with Hidden Reference and Anchors* procedure (MUSHRA, ITU-R BS.1534) [1, 2, 14] to evaluate each scenario, using four tests stimuli (binaural reference, B-format, our approach with foreground only, our approach with background/foreground segmentation) and a hidden reference. We also provided one of our 8 monophonic recordings and the omnidirectional (W) component of the B-format recordings as anchors, resulting in a total of 7 signals to compare. Corresponding test stimuli are available at the following URL: http://www-sop.inria.fr/reves/projects/aes30. Test stimuli were presented over *Sennheiser HD600* headphones. Monaural anchor signals were presented at both ears.

Five subjects, aged 23 to 40 and reporting normal hearing, volunteered for this evaluation. They were asked to primarily focus on the spatial aspects of the sounds, paying particular attention to the position of the sources. Since the recordings were made with different microphones, we asked them to avoid specific judgments comparing the general timbre of the recordings. However, the subjects were instructed to keep track of any artefact compromising the quality of the reproduction. Their comments were gathered during a short post-screening interview. Subjects were instructed to rank the signals on a continuous [0,100] quality scale and give the highest possible score to the signal closest to the reference. They were also instructed to give the lowest possible score to the signal with the worst *spatial degradation* relative to the reference.

3.2. Results

Figures 7 and 8 summarize the results of this study. The subjects were able to identify the hidden reference and it received a maximal score in all test cases. In most cases, our approach was rated higher than B-format recordings in terms of quality of spatial reproduction. This is particularly true for the foreground-only approach which does not smooth the spatial cues and obtains a very high score. However, the subjects reported artefacts due to subbands whose localization varies rapidly through time, which limits the applicability of the approach in noisier environments. Our approach including background/foreground separation leads to smoother spatial cues since the low order background signal may mask the foreground signal. Hence, it was rated only slightly better than the B-format recordings. Subjects did not report specific artefacts with this approach, showing an improved signal quality. As could be expected, the monophonic anchors received the lowest scores. However, we can note that in some of our test cases, they

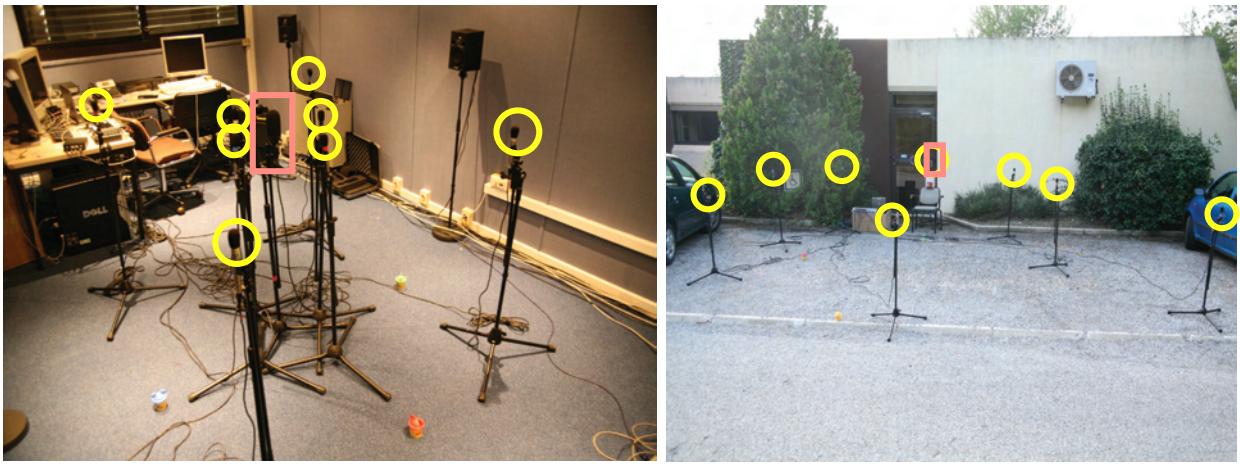


Fig. 6: Example recording setups. We used 8 omnidirectional microphones (circled in yellow) to capture the auditory scene as well as a *Soundfield* microphone (highlighted with a light red square) to simultaneously record a B-format version. A binaural recording using microphones placed in the ears of a subject provided a reference recording in each test case.

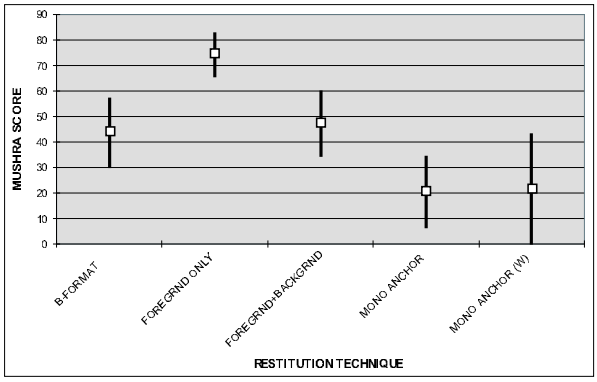


Fig. 7: Average MUSHRA scores and 95% confidence intervals for all subjects and all scenarios.

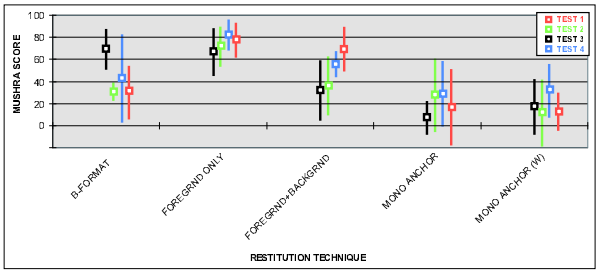


Fig. 8: Average MUSHRA scores and 95% confidence intervals for all subjects in each of our 4 test scenarios.

received scores very close to the B-format reproduction. This is probably due to the low spatial resolution of B-format but could also arise from a non-optimal HRTF-based decoding.

Looking at the various test-cases in more detail, Figure 8 highlights a significantly different behavior for the indoor scenario (TEST#3). In this case, very little background sound was present, hence our approach based on background and foreground separation did not lead to any improvement and, in fact, resulted in a degraded spatial impression. The B-format reproduction, however, obtained significantly better scores in this case, probably due to the favorable configuration of the three speakers (one in front, one to the left, and one to the right).

3.3. Discussion

In terms of audio quality, feedback from the subjects of the tests shows that our improved algorithm outperforms the previous foreground-only solution. This is of course due to the smoothly varying background and more robust foreground estimates. However, our proposed approach appears less convincing in terms of localization accuracy. Significant parts of the foreground sounds can still be present in the background component and will be spatialized using a different strategy. The resulting

Fig. 9: Recording setup used for the seashore recordings.

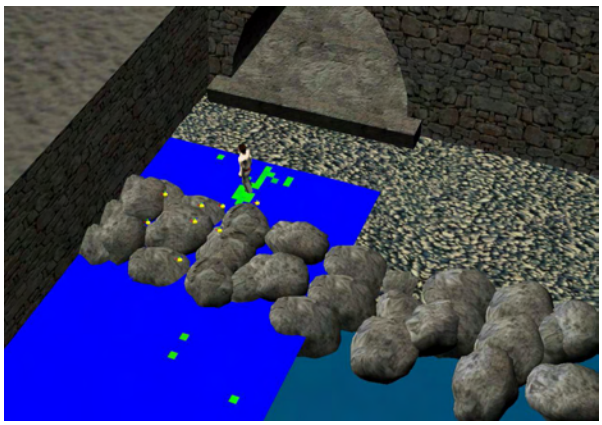


Fig. 10: Example virtual reconstruction of a seashore with walking pedestrian. Yellow spheres correspond to the locations of the microphones used for recording.

blend tends to blur out the localization cues leading to a poorer spatial impression. Improving the quality of the segmentation would probably lead to better results. Another possibility would be to use energy and not only time-differences of arrival to extract possible localization information for the background component.

We used a small number of frequency subbands in our tests which can challenge our time-frequency orthogonality assumption resulting in noisier position estimates for the foreground component. However, we obtained less convincing results with an increased number of frequency subbands due to less accurate correlation estimates for narrower subbands signals.

We do not currently model sources "at infinity", which may appear in the background but also in the foreground component. Our position estimation can return erroneous position estimates in this case due to the limited extent of our position histogram. This could also explain the perceived degradation of spatial cues compared to the reference. Explicit detection of far-field sources is a component we are planning to add in the near future. Finally, non-individualized HRTF processing could also be a major cause of spatial degradation. Running the test with head-tracking and individualized HRTFs might lead to improved results.

4. APPLICATIONS

Our approach can lead to spatial audio coding applications for live audio footage in a way similar to [28, 13, 29, 6], but it also offers novel decoding/authoring capabilities not available with previous techniques such as *free-viewpoint* walkthroughs. Figure 10 illustrates the virtual reconstruction of a seashore scene with a pedestrian walking on a pebble beach recorded with the setup shown in Figure 9. A spatial energy map is overlaid, highlighting the location of foreground time-frequency atoms. Note how the position of the footsteps sounds is well reconstructed by our approach. The sound of sea waves hitting the rocks on the shore is mostly captured by the background component (see also Figure 5). Please, visit the web pages mentioned in Sections 1 and 3 for example audio files and videos.

Spatial re-synthesis with free-moving listener

Our approach allows for a "free-viewpoint" spatial audio rendering of the acquired soundscapes. As the virtual listener moves throughout the scene, the foreground component is rendered using a collection of point sources corresponding to each time-frequency atom, as described in section 2.3. The background component is simply rotated based on the current orientation of the listener in order to provide a consistent rendering. Our representation encodes spatial cues in world space and can thus be rendered on a variety of reproduction setups (headphones, multichannel, *etc.*).

Background/foreground editing

Our two-layer model allows for independent control of the background and foreground components. Their overall level can be adjusted globally or locally, for instance to attenuate foreground sounds with local virtual occluders while preserving the background. The foreground

events can also be copied and pasted over a new background ambiance.

Re-rendering with various microphones

Finally, the microphones used for the analysis process can be different from the one used for re-rendering. For instance, it is possible to use any directional microphone to get a combined effect of spatial rendering and beamforming.

5. CONCLUSION

We presented an approach to convert field recordings into a structured representation suitable for generic 3D audio processing and integration with 2D or 3D visual content. It applies both to outdoor environments or indoor environments with limited reverberation, provides a compact encoding of the spatial auditory cues and captures propagation and reverberation effects which would be very difficult to render with the same level of realism using traditional virtual acoustics simulations.

Perceptual comparisons with reference binaural and B-format recordings showed that our approach outperforms B-format recordings and can get close to reference binaural recordings when all time-frequency atoms are rendered as foreground point sources. However, artefacts due to background noise lead to reduced signal quality. An alternative solution was proposed based on the explicit segmentation of stationary "background noise" and non-stationary "foreground events". While the signal quality is significantly improved when re-rendering, spatial cues were perceived to be degraded, probably due to non-optimal background separation.

In the future, we would like to improve on our background/foreground segmentation approach, possibly based on auditory *saliency* models [17] or taking advantage of the signals from all microphones. Alternative sparse representations of the signals [22, 21] could also be explored in order to improve our approach. Further comparisons to other sound-field acquisition techniques, for instance based on high-order spherical harmonic encoding [3, 24], Fourier-Bessel decomposition [19, 20] or directional audio coding [27, 28] would also be of primary interest to evaluate the quality vs. flexibility/applicability tradeoffs of the various approaches. We believe our approach opens many novel perspectives for interactive spatial audio rendering or off-line post-production environments, for example to complement image based rendering techniques or free-viewpoint video.

6. ACKNOWLEDGMENTS

This research was made possible by a grant from the *région PACA* and was partially funded by the RNTL project OPERA (http://www-sop.inria.fr/reves/OPERA).

7. REFERENCES

[1] EBU subjective listening tests on internet audio codecs. *EBU TECHNICAL REVIEW, European Broadcast Union (EBU)*, june 2000.

[2] EBU subjective listening tests on low-bitrate audio codecs. *Technical report 3296, European Broadcast Union (EBU), Projet Group B/AIM*, june 2003.

[3] T. Abhayapala and D. Ward. Theory and design of high order sound field microphones using spherical microphone array. *Proceedings of Intl. Conf. on Acoustics, Speech and Signal Processing*, 2002.

[4] F. Baumgarte and C. Faller. Binaural cue coding - part I: Psychoacoustic fundamentals and design principles. *IEEE Transaction on Speech and Audio Processing*, 11(6), 2003.

[5] D. R. Begault. *3D Sound For Virtual Reality and Multimedia.* Academic Press, Inc., 1994.

[6] J. Breebaart, J. Herre, C. Faller, J. Rödén, F. Myburg, S. Disch, H. Purnhagen, G. Hotho, M. Neusinger, K. Kjörling, and W. Oomen. MPEG spatial audio coding/MPEG surround: Overview and current status. *Proc. 119th AES Convention, New York, USA. Preprint 6599*, October 2005.

[7] J. Chen, J. Benesty, and Y. A. Huang. Time delay estimation in room acoustic environments: An overview. *EURASIP Journal on Applied Signal Processing*, 2006.

[8] C. Choi. Real-time binaural blind source separation. *Proc. of the 4th Intl. Symp. on Independant Component Analysis and Blind Source Separation (ICA2003), Nara, Japan, april*, 2003.

[9] Y. Ephraim and D. Malah. Speech enhancement using a- minimum mean-square error short-time spectral amplitude estimator. *IEEE Trans. on Acoustics, Speech and Signal*, ASSP-32(6):1109–1121, December 1984.

[10] C. Faller and F. Baumgarte. Binaural cue coding - part II: Schemes and applications. *IEEE Transaction on Speech and Audio Processing*, 11(6), 2003.

[11] O. Faugeras. *Three-dimensional computer vision: a geometric viewpoint*. MIT Press, 1993.

[12] E. Gallo, N. Tsingos, and G. Lemaitre. 3D-Audio matting, post-editing and re-rendering from field recordings. *EURASIP Journal on Applied Signal Processing, special issue on Spatial Sound and Virtual Acoustics*, 2007.

[13] M. Goodwin and J.-M. Jot. Analysis and synthesis for universal spatial audio coding. *In 121th AES Convention, San Francisco, USA. Preprint 6874*, 2006.

[14] International Telecom. Union. Method for the subjective assessment of intermediate quality level of coding systems. *Recommendation ITU-R BS.1534-1*, 2001-2003.

[15] J.-M. Jot, V. Larcher, and J.-M. Pernaux. A comparative study of 3D audio encoding and rendering techniques. *Proceedings of the AES 16th international conference, Spatial sound reproduction, Rovaniemi, Finland*, april 1999.

[16] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transaction of the ASME-Journal of Basic Engineering 82 (Series D)*, pages 35–45, 1960.

[17] C. Kayser, C. Petkov, M. Lippert, and N. Logothetis. Mechanisms for allocating auditory attention: An auditory saliency map. *Current Biology*, 15:1943–1947, Nov. 2005.

[18] C. Knapp and G. C. C. and. The generalized correlation method for estimation of time delay. *IEEE Transaction on Acoustics, Speech and Signal Processing*, 4(4):320–327, 1976.

[19] A. Laborie, R. Bruno, and S. Montoya. A new comprehensive approach of surround sound recording. *Proc. 114th convention of the Audio Engineering Society, preprint 5717*, 2003.

[20] A. Laborie, R. Bruno, and S. Montoya. High spatial resolution multi-channel recording. *Proc. 116th convention of the Audio Engineering Society, preprint 6116*, 2004.

[21] M. S. Lewicki and T. J. Sejnowski. Learning overcomplete representations. *Neural Computation*, 12(2):337–365, 2000.

[22] S. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993.

[23] J. Merimaa. Applications of a 3D microphone array. *112th AES convention, preprint 5501*, 2002.

[24] J. Meyer and G. Elko. Spherical microphone arrays for 3d sound recording. *chap. 2 in Audio Signal Processing for next-generation multimedia communication systems, Eds. Yiteng (Arden) Huang and Jacob Benesty, Bosten, Kluwer Academic Publisher*, 2004.

[25] J. Meyer and G. Elko. *Spherical microphone arrays for 3D sound recording, chap. 2 in Audio Signal Processing for next-generation multimedia communication systems, Eds. Yiteng (Arden) Huang and Jacob Benesty, Bosten, Kluwer Academic Publisher*. 2004.

[26] R. D. Patterson and B. C. J. Moore. *Auditory filters and excitation patterns as representations of auditory frequency selectivity, in Frequency Selectivity in Hearing, Academic Press, London*, pages 123–177. 1986.

[27] V. Pulkki. Directional audio coding in spatial sound reproduction and stereo upmixing. *Proc. of the AES 28th Int. Conf, Pitea, Sweden*, June 2006.

[28] V. Pulkki and C. Faller. Directional audio coding: Filterbank and stft-based design. *In 120th AES Convention, Paris, France, Preprint 6658.*, May 20-23 2006.

[29] V. Pulkki and J. Merimaa. Spatial impulse response rendering. *Proc. of the 7th Intl. Conf. on Digital Audio Effects (DAFX'04), Naples, Italy*, 2004.

[30] R. Radke and S. Rickard. Audio interpolation. *In the Audio Engineering Society 22nd International Conference on Virtual, Synthetic and Entertainment Audio (AES'22), Espoo, Finland*, pages 51–57, 2002.

[31] S. Rickard. Sparse sources are separated sources. *Proceedings of the 16th Annual European Signal Processing Conference, Florence, Italy*, 2006.

[32] H. Samet. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley, 1990.

[33] L. Savioja, J. Huopaniemi, T. Lokki, and R. Väänänen. Creating interactive virtual acoustic environments. *J. of the Audio Engineering Society*, 47(9):675–705, Sept. 1999.

[34] SOUNDFIELD. http://www.soundfield.com.

[35] R. Streicher. The decca tree - it's not just for stereo anymore, http://www.wesdooley.com/pdf/surround sound decca tree-urtext.pdf., 2003.

[36] H. Traunmüller. Analytical expressions for the tonotopic sensory scale. *The Journal of the Acoustical Society of America*, 88:97–100, July 1990.

[37] E. Vincent, C. Févotte, R. Gribonval, X. Rodet, É. L. Carpentier, L. Benaroya, A. Rödel, and F. Bimbot. A tentative typologie of audio source separation tasks. *Proc. of the 4th Intl. Symposium on Independent Component Analysis and Blind Signal Separation (ICA2003), Nara, Japan*, 2003.

[38] Ö. Yilmaz and S. Rickard. Blind separation of speech mixtures via time-frequency masking. *IEEE Transactions on Signal Processing*, 52(7), July 2004.

Chapter 7

Conclusion

In this work, we proposed a number of models and algorithms for interactive audio rendering. We proposed original solutions for interactive and accurate physically-based simulations, fast signal processing and 3D audio rendering using perceptual principles and authoring of realistic scenes from spatial recordings.

In particular, we showed that it is possible to peform interactive geometrical simulation of early sound specular reflections and edge diffraction using an improved beam-tracing approach. We also introduced a shadow-region approximation for egde diffraction that allows for treating large environments by limiting the spatial extent of diffracted beams. We proposed a solution beyond the classical sound-ray formalism by using the Kirchhoff approximation. This approach can be seen as a hybrid geometrical/wave solution that can handle very detailed geometry. It is currently limited to first-order scattering and still shares some of the low-frequency limitations of classical geometrical acoustics. However, we believe that it eventually offers more potential for improving accuracy than ray-based techniques. We also explored how programmable graphics hardware can be used both for massive audio signal processing but also to sample and compute complex surface integrals, as required by the Kirchhoff approximation. Our results show that such hardware architectures are very well suited to acoustics simulations.

Our second contribution proposed an alternative solution to massive brute-force calculation to auralize complex virtual auditory scenes. By exploiting limitations of our auditory perception, we proposed simple but efficient algorithms to progressively simplify and render the original scene. A departure from previous approaches is to use compact, pre-computed, signal descriptors to drive the simplifications, hence making the algorithms "content-aware". Similar to hidden-surface removal in computer graphics, we introduced a concurrent masking algorithm that dynamically re-evaluates inaudible sounds at every time-frame of the simulation. Inaudible sounds need not be processed, nor even streamed/decompressed. We also demonstrated the usability of this approach for bandwith management in a spatialized voice-over-IP application. We extended this approach by pre-computing a progressive representation of our signals which can then be reconstructed and processed at a variable "bit-rate" based on their relative importance to the final mix. These approaches build upon perceptual compression schemes but perform "inter-sound" masking and budget allocation. Another algorithm was proposed to simplify the spatial auditory cues present in the original scene. Assuming that our scene is modeled as a collection of 3D point sources, we proposed a dynamic clustering strategy that groups nearby sources and build a single "impostor" source for each cluster. 3D audio processing is then amortized over clusters of sources rather than being performed individually for each source. These approaches have been shown to be very effective while largely preserving the perceived rendering quality.

As virtual auditory scenes become more and more complex, the traditional approach of authoring

3D audio content using large collections of point sources does not scale. An alternative is to directly use recordings embedding the spatial information of the scene so that they can be re-rendered from various listening points and allow for realtime post-processing effects. Our third contribution is a first solution to implement such an approach. We demonstrated how to extract spatial structure from multi-point recordings of a real-world auditory scene, estimating position of the sound sources and separation of background and foreground components. This step can be seen as a conversion of the original set of recordings into a set of point sound sources in 3D space. This additional spatial information is represented in world-space thanks to a spatial calibration step of the microphone array and stored as a side information. It allows for manipulating the scene interactively and independently of the reproduction setup, as a post-processing step and shares some similarities with recent Spatial Audio Object Coding (SAOC) work [HD07]. We demonstrated and discussed possible applications to virtual walkthroughs in the recording and repositionning or altering some of the sound sources. Several such recordings can also be merged together in a spatially consistent way. While the quality of the reconstructed results is not yet compatible with professional production environments, we believe this solution could offer a way to automatically author complex, spatially extended, sound sources and auditory scenes from recordings. The resulting clouds of point sources could be easily rendered using the efficient approaches offered by our previous contribution.

7.1 Perspectives

Obtaining accurate simulations at interactive rates is likely to remain and enduring endeavour of physically-based acoustics. Being able to perform more accurate simulations at faster rates will allow to solve problems not only related to room acoustics but also to 3D sound reproduction. For instance, a number of approaches have attempted to use numerical simulation in order to individualize 3D audio rendering over headphones. However, the cost of current finite element simulations prohibits such solutions from being used on a large scale. With the development of mobile entertainment and the playback of surround sound over portable media players, this issue is likely to become a key challenge in 3D audio rendering.

For a number of important applications not requiring physical accuracy, we believe perceptual solutions are a compelling alternative. For instance, artifical reverberators based on recursive filtering have been successful in a a number of mass-market applications, including video games. Some of our perceptually-based auralisation techniques are already in use in commercial video games which makes us confident that perceptually-based audio rendering is a promising area for future research. Distributed audio streaming, such as massive spatialized “chat” over IP networks can also be envisioned as a future driving element for perceptually-based processing approaches. In this context, they can be used to optimize bandwith but also to reduce computational load on conferencing bridges.

Authoring of 3D audio content has never been subject to intensive research when compared to its visual counterpart. While techniques are available for 3D audio rendering and recording, authoring environments and interfaces could be drastically improved. In particular, there is a need of rendering approaches and authoring tools to bridge the gap between scenes modeled by monophonic point sources and surround sound recordings. The former provide full flexibility at the expense of a tedious modeling step. The latter allow for direct recording of spatial audio content but offer limited post-editing capabilities, if any. More complex spatial audio recording approaches have been proposed to date but do not seem to be considered as viable solutions by the production industry. We believe that alternative “structure from recording” approaches, capable of converting multi-channel 3D audio recordings into clouds of point sources, would be tools of enormous interest to the interactive media industry.

Finally a key element, not covered by our contributions, is interactive content generation and in

particular sound synthesis for virtual environment applications. Traditionnaly relying on pre-recorded sounds, we believe that most 3D rendering pipelines would benefit from physically-based synthesis. As video games and simulators include more and more sophisticated real-time physics engines, all the required data is available to directly synthesize compelling sounds to picture. Contact sounds between objects are obvious candidates for synthesis and a number of solutions are already available. We are currently experimenting with such techniques and believe that some of our contributions to scalable processing have promising applications for physically-based synthesis of contact sounds.

Bibliography

[AAD99] Carlos Avendano, V. Ralph Algazi et Richard O. Duda. – A head-and-torso model for low-frequency binaural elevation effects. *In : Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. – New Paltz, New York, octobre 1999.

[AAD01] V. Ralph Algazi, Carlos Avendano et Richard O. Duda. – Estimation of a spherical-head model from anthropometry. *J. of the Audio Engineering Society*, 2001.

[AB79] J.B. Allen et D.A. Berkley. – Image method for efficiently simulating small room acoustics. *J. of the Acoustical Society of America*, vol. 65, n° 4, 1979.

[AB04] D. Alais et D. Burr. – The ventriloquism effect results from near-optimal bimodal integration. *Current Biology*, vol. 14, pp. 257–262, 2004.

[AC01] Daniel G. Aliaga et Ingrid Carlbom. – Plenoptic stitching: a scalable method for reconstructing 3d interactive walk throughs. *In : SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. pp. 443–450. – New York, NY, USA, 2001.

[ADT04] V.R. Algazi, R.O. Duda et D.M. Thomson. – Motion-tracked binaural sound. *J. of the Audio Engineering Society*, vol. 52, n° 11, pp. 1142–1156, novembre 2004.

[AE03] Marios Athineos et Daniel P.W. Ellis. – Sound texture modelling with linear prediction in both time and frequency domains. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'03)*, 2003.

[AM99] L. Aveneau et M. Meriaux. – Rendering polygonal scenes with diffraction account. *Seventh International Conference in Central Europe on Computer Graphics and Visualization (Winter School on Computer Graphics)*, February 1999. – ISBN 80-7082-490-5. Held in University of West Bohemia, Plzen, Czech Republic, 10-14 February 1999.

[Ama84] John Amanatides. – Ray tracing with cones. *ACM Computer Graphics, SIGGRAPH'84 Proceedings*, vol. 18, n° 3, pp. 129–135, juillet 1984.

[AS83] H. Alrutz et M.R. Schroeder. – A fast hadamard transform method for evaluation of measurements using pseudorandoom test signals. pp. 235–238.

[aud] AUDIS HRTF database. available from the european acoustics association (EAA). http://www.euracoustics.org.

[AV02] T. Ajdler et M. Vetterli. – The plenacoustic function and its sampling. *Proc. of the 1st Benelux Workshop on Model-based processing and coding of audio (MPCA2002), Leuven, Belgium*, novembre 2002.

[Ave03] C. Avendano. – Frequency-domain source identification and manipulation in stereo mixes for enhancement, suppresssion and re-panning applications. *Proc. of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA2003), New Paltz, NY, USA*, octobre 2003.

[AW02] T.D. Abhayapala et D.B. Ward. – Theory and design of high order sound field microphones using spherical microphone array. *Proceedings of Intl. Conf. on Acoustics, Speech and Signal Processing*, 2002.

[AWBW05] M. Allman-Ward, M.P. Balaam et R. Williams. – Source decomposition for vehicle sound simulation. *available from www.mts.com/nvd/pdf/source_decomp4veh_soundsim.pdf*, 2005.

[BBM+01] Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler et Michael Cohen. – Unstructured lumigraph rendering. *Proc. of ACM SIGGRAPH*, 2001.

[BD98] C. Philip Brown et Richard O. Duda. – A structural model for binaural sound synthesis. *IEEE Trans. on Speech and Audio Processing*, vol. 6, n° 5, septembre 1998.

[BdVV93] A.J. Berkhout, D. de Vries et P. Vogel. – Acoustic control by wave field synthesis. *J. of the Acoustical Society of America*, vol. 93, n° 5, pp. 2764–2778, may 1993.

[Beg94] Durand R. Begault. – *3D Sound for Virtual Reality and Multimedia*. – Academic Press Professional, 1994.

[BF03] Frank Baumgarte et Christof Faller. – Binaural cue coding - part I: Psychoacoustic fundamentals and design principles. *IEEE Trans. on Speech and Audio Proc*, vol. 11, n° 6, 2003.

[BFH+04] I. Buck, T. Foley, D. Horn, J. Sugerman et P. Hanrahan. – Brook for GPUs: Stream computing on graphics hardware. *ACM Transactions on Graphics, Proceedings of SIGGRAPH 2004*, août 2004.

[BHF+05] J. Breebaart, J. Herre, C. Faller, J. Rödén, F. Myburg, S. Disch, H. Purnhagen, G. Hotho, M. Neusinger, K. Kjörling et W. Oomen. – MPEG spatial audio coding/MPEG surround: Overview and current status. *Proc. 119th AES Convention, New York, USA. Preprint 6599*, October 2005.

[Bil06] Frans A. Bilsen. – Repetition pitch glide from the step pyramid at Chichen Itza. *J. of the Acoustical Society of America*, no120, p. 594, 2006.

[BJLW+99] Ziv Bar-Joseph, Dani Lischinski, Michael Werman, Ran El-Yanniv et Shlomo Dubnov. – Granular synthesis of sound textures using statistical learning. *International Computer Music Conference (ICMC'99)*, 1999.

[BKW04] A. Blum, B.F.G. Katz et O. Warusfel. – Eliciting adaptation to non-individual hrtf spectral cues with multi-modal training. *Proc. of CFA/DAGA, Strasbourg, France*, mars 2004.

[Bla97] J. Blauert. – *Spatial Hearing : The Psychophysics of Human Sound Localization*. – M.I.T. Press, Cambridge, MA, 1997.

[Ble01] B. Blesser. – An interdisciplinary integration of reverberation. *J. of the Audio Engineering Society*, vol. 49, n° 10, pp. 867–903, 2001.

[Bor84] J. Borish. – Extension of the image model to arbitrary polyhedra. *J. of the Acoustical Society of America*, vol. 75, n° 6, 1984.

[Bre90] A.S. Bregman. – *Auditory Scene Analysis, The perceptual organization of sound*. – The MIT Press, 1990.

[BSK05] Douglas S. Brungart, Brian D. Simpson et Alexander J. Kordik. – Localization in the presence of multiple simultaneous sounds. *Acta Acustica united with Acustica*, vol. 91, pp. 471–479(9), May/June 2005.

[BvSJC05] Virginia Best, Andre van Schaik, Craig Jin et Simon Carlile. – Auditory spatial perception with sources overlapping in frequency and time. *Acta Acustica united with Acustica*, vol. 91, pp. 421–428(8), May/June 2005.

[BW02] A. Baskind et O. Warusfel. – Methods for blind computational estimation of perceptual attributes of room acoustics. *proceedings of the AES 22nd Intl. Conf. on virtual, synthetic and entertainment audio, Espoo, Finland*, juin 2002.

[BWA01] D. Begault, E. Wenzel et M. Anderson. – Direct comparison of the impact of head-tracking, reverberation, and individualized head-related transfer functions on the spatial perception of a virtual speech source. *J. of the Audio Engineering Society*, vol. 49, n° 10, pp. 904–916, 2001.

[BZ03] Matthias Baeck et Udo Zölzer. – Real-time implementation of a source separation algorithm. *Proc. of Digital Audio Effects Conference (DAFX'03)*, 2003.

[CC95] J.H. Chuang et S.A. Cheng. – Computing caustic effects by backward beam tracing. *The Visual Computer*, vol. 11, n° 3, pp. 156–166, 1995.

[CCC87] Robert L. Cook, Loren Carpenter et Edwin Catmull. – The reyes image rendering architecture. *SIGGRAPH Comput. Graph.*, vol. 21, n° 4, pp. 95–102, New York, NY, USA, 1987.

[CI90] J.S. Chen et A. Ishimaru. – Numerical simulation of the second-order Kirchhoff approximation from very rough surfaces and a study of backscattering enhancement. *J. Acous. Soc. of America*, no4, pp. 1846–1850, Oct 1990.

[cip] CIPIC HRTF database. available from:. http://interface.cipic.ucdavis.edu/CIL_html/CIL_HRTF_database.htm.

[CL93] T.J. Cox et Y.W. Lam. – Evaluations of methods for predicting the scattering from simple rigid panels. *Applied Acoustics*, vol. 40, pp. 123–140, 1993.

[CM82] L. Cremer et H.A. Müller. – *Principles and Applications of Room Acoustics, Vols 1 and 2; translated by T.J. Shultz.* – Applied Science Publishers, London, 1982.

[COM98] Jonathan Cohen, Marc Olano et Dinesh Manocha. – Appearance-preserving simplification. *In : SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. pp. 115–122. – New York, NY, USA, 1998.

[Coo02] P.R. Cook. – *Real Sound Synthesis for Interactive Applications*. – AK Peters, 2002.

[CVH95] J. Chen, B.D. Van Veen et K.E. Hecox. – A spatial feature extraction and regularization model for the head-related transfer function. *J. of the Acoustical Society of America*, vol. 97, pp. 439–452, janvier 1995.

[CW93a] S.E. Chen et L. Williams. – View interpolation for image synthesis. *Computer Graphics*, vol. 27, n° Annual Conference Series, Proc. of ACM SIGGRAPH93, pp. 279–288, 1993.

[CW93b] Michael F. Cohen et John R. Wallace. – *Radiosity and Realistic Image Synthesis*. – Academic Press Professional, 1993.

[DAA99] Richard O. Duda, Carlos Avendano et V. Ralph Algazi. – An adaptable ellipsoidal head model for the interaural time difference. *In : Proc. IEEE Int. Conference on Acoustics Speech and Signal Processing (ICASSP'99)*, pp. II:965–968.

[Dal96] Bengt-Inge L. Dalenbäck. – Room acoustic prediction based on a unified treatment of diffuse and specular reflection. *J. of the Acoustical Society of America*, vol. 100, n° 2, pp. 899–909, août 1996.

[Dan00] Jerôme Daniel. – *Repr'esentation de champs acoustiques, application à la transmission et à la reproduction de scènes sonores complexes dans un contexte multimédia*. – Thèse de doctorat, Université de Paris VI, juillet 2000.

[DBJEY+02] Shlomo Dubnov, Ziv Bar-Joseph, Ran El-Yanniv, Dani Lischinski et Michael Werman. – Synthesis of sound textures by learning and resampling of wavelet trees. *IEEE Computer Graphics and Applications*, 2002.

[DCH00] Myriam Desainte-Catherine et Pierre Hanna. – Statistical approach for sound modeling. *In : Proc. of Digital Audio Effects Conference (DAFX'00),Verona, Italy*, pp. 91–96.

[DDBL04] Nico F. Declercq, Joris Degrieck, Rudy Briers et Oswald Leroy. – A theoretical study of special acoustic effects caused by the staircase of the El Castillo pyramid at the maya ruins of Chichen-Itza in Mexico. *J. of the Acoustical Society of America*, no116, p. 3328, 2004.

[DDS02] D. Darlington, L. Daudet et M. Sandler. – Digital audio effects in the wavelet domain. *In : Proceedings of COST-G6 Conference on Digital Audio Effects, DAFX2002, Hamburg, Germany*.

[DFMM99] G. Dickins, M. Flax, A. McKeag et D. McGrath. – Optimal 3D-speaker panning. *Proceedings of the AES 16th international conference, Spatial sound reproduction, Rovaniemi, Finland*, pp. 421–426, april 1999.

[Dir04] Direct Sound 3D. – Direct X homepage, Microsoft©, 2004. http://www.microsoft.com/windows/directx/default.asp.

[DJ00] L. Dahl et J.M. Jot. – A reverberator based on absorbent all-pass filters. *Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-00), Verona, Italy*, décembre 2000.

[DKW82] N. Dadoun, D.G. Kirkpatrick et J.P. Walsh. – Hierarchical approaches to hidden surface intersection testing. *Graphics Interface '82*, pp. 49–56, May 1982.

[DKW85] N. Dadoun, D.G. Kirkpatrick et J.P. Walsh. – The geometry of beam tracing. *Proceedings of the Symposium on Computational Geometry*, pp. 55–71, June 1985.

[DM04] J. Daniel et S. Moreau. – Further study of sound field coding with higher order Ambisonics. *AES 116th convention, preprint 6017*, 2004.

[Do04] M.N. Do. – Toward sound-based synthesis: the far-field case. *Proc. of IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), Montreal, Canada*, mai 2004.

[Doe04] K. Van Den Doel. – Physically-based models for liquid sounds. *Proceedings of Intl. Conf. on Auditory Display*, Sydney, Australia, July 2004.

[DP98] K. Van Den Doel et D.K. Pai. – The sound of physical shapes. *Presence*, vol. 7, n° 4, pp. 382–395, 1998.

[DS96] S. Van Duyne et J.O. Smith. – The 3D tetrahedral digital waveguide mesh with musical applications. *Proceedings of Intl. Computer Music Conf.*, vol. (ICMC96), pp. 9–16, août 1996.

[DS03] Agostino Di-Scipio. – Synthesis of environmental sound textures by iterated non linear fonctions. *Proc. of Digital Audio Effects Conference (DAFX'03)*, 2003.

[DS05] C. Dachsbacher et M. Stamminger. – Reflective shadow map. *Proceedings of I3D'05*, 2005.

[DYN03] Y. Dobashi, T. Yamamoto et T. Nishita. – Real-time rendering of aerodynamic sound using sound textures based on computational fluid dynamics. *ACM Transactions on Graphics*, vol. 22, n° 3, pp. 732–740, août 2003. – (Proceedings of ACM SIGGRAPH 2003).

[DYN04] Y. Dobashi, T. Yamamoto et T. Nishita. – Synthesizing sound from turbulent field using sound textures for interactive fluid simulation. *Computer Graphics Forum (Proc. EUROGRAPHICS 2004)*, vol. 23, n° 3, pp. 539–546, 2004.

[EAX04] EAX. – Environmental audio extensions 4.0, Creative©, 2004. http://www.soundblaster.com/eaudio.

[EB00] J. Eyre et J. Bier. – The evolution of DSP processors. *IEEE Signal Processing Magazine*, 2000. – See also http://www.bdti.com/.

[EDS01] J.J. Embrechts, D.Archambeau et G.B. Stan. – Determination of the scattering coefficient of random rough diffusing surfaces for room acoustics applications. *Acta Acustica united with Acustica*, vol. 87, pp. 482–494, June 2001.

[ELD91] Judy Edworthy, Sarah Loxley et Ian Dennis. – Improving auditory warning design: Relationship between warning sound parameters and perceived urgency. *Human Factors*, vol. 33, n° 2, pp. 205–231, 1991.

[EM84] Y. Ephraim et D. Malah. – Speech enhancement using a- minimum mean-square error short-time spectral amplitude estimator. *IEEE Trans. on Acoustics, Speech and Signal*, vol. ASSP-32, n° 6, pp. 1109–1121, December 1984.

[Emb00] J.J. Embrechts. – Simulation of first and second-order scattering by rough surfaces with a sound-ray formalism. *J. of Sound and Vibration*, vol. 229, n° 1, pp. 65–87, June 2000.

[Eme95] Marc Emerit. – *Simulation Binaurale de l'Acoustique de Salles de Concert*. – Thèse de doctorat, Centre Scientifique et Technique du Bâtiment, 1995.

[Fau93] O. Faugeras. – *Three-Dimensional Computer Vision: A Geometric Viewpoint*. – The MIT Press, Cambridge, Mass., 1993.

[FB03] Christof Faller et Frank Baumgarte. – Binaural cue coding - part II: Schemes and applications. *IEEE Trans. on Speech and Audio Proc*, vol. 11, n° 6, 2003.

[FCE+98] T. Funkhouser, I. Carlbom, G. Elko, G. Pingali, M. Sondhi et J. West. – A beam tracing approach to acoustic modeling for interactive virtual environments. *ACM Computer Graphics, SIGGRAPH'98 Proceedings*, pp. 21–32, juillet 1998.

[FHB97] H. Fouad, J.K. Hahn et J.A. Ballas. – Perceptually based scheduling algorithms for real-time synthesis of complex sonic environments. *proceedings of the 1997 International Conference on Auditory Display (ICAD'97), Xerox Palo Alto Research Center, Palo Alto, USA*, 1997.

[FHLB99] P. Filippi, D. Habault, J.P. Lefevre et A. Bergassoli. – *Acoustics, basic physics, theory and methods*. – Academic Press, 1999.

[FJT02] T. Funkhouser, J.M. Jot et N. Tsingos. – Sounds good to me ! computational sound for graphics, vr, and interactive systems. *Siggraph 2002 course #45*, 2002.

[FM04] Christof Faller et Juha Merimaa. – Source localization in complex listening situations: Selection of binaural cues based on interaural coherence,. *J. Acoust. Soc. Am*, vol. 116, n° 5, Nov 2004.

[FMC99] T. Funkhouser, P. Min et I. Carlbom. – Real-time acoustic modeling for distributed virtual environments. *ACM Computer Graphics, SIGGRAPH'99 Proceedings*, pp. 365–374, août 1999.

[FMO] FMOD music and sound effects system. http://www.fmod.org.

[For96] Steve Fortune. – *Algorithms for Prediction of Indoor Radio Propagation*. – Rapport technique n° Document 11274-960117-03TM, Bell Laboratories, 1996.

[For99] S.J. Fortune. – Topological beam tracing. *In : Proc. 15th ACM Symposium on Computational Geometry*, pp. 59–68.

[FST92] Thomas Funkhouser, Carlo H. Sequin et Seth J. Teller. – Management of large amounts of data in interactive building walkthroughs. *ACM Computer Graphics (1992 SIGGRAPH Symposium on Interactive 3D Graphics)*, pp. 11–20, March 1992.

[FU97] A. Farina et E. Ugolotti. – Subjective comparison of different car audio systems by the auralization technique.

[Fuj88] Akira Fujomoto. – Turbo beam tracing - A physically accurate lighting simulation environment. *Knowledge Based Image Computing Systems*, pp. 1–5, May 1988.

[FVFH90] Foley, VanDam, Feiner et Hughes. – *Computer graphics, principles and practice*. – Addison Wesley, 1990.

[Gar95] Bill Gardner. – *Transaural 3D audio*. – Rapport technique n° 342, M.I.T. Media Lab Perceptual Computing, juillet 1995.

[Ger85] M.A. Gerzon. – Ambisonics in multichannel broadcasting and video. *J. of the Audio Engineering Society*, vol. 33, n° 11, pp. 859–871, 1985.

[GGSC96] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski et Michael F. Cohen. – The lumigraph. *In : SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. pp. 43–54. – New York, NY, USA, 1996.

[GH98] G. Ghazanfarpour et J. Marc Hasenfratz. – A beam tracing with precise antialiasing for polyhedral scenes. *Computers and Graphics*, vol. 22, n° 1, 1998.

[GJ06] Michael Goodwin et Jean-Marc Jot. – Analysis and synthesis for universal spatial audio coding. *In 121th AES Convention, San Francisco, USA. Preprint 6874*, 2006.

[GK96] E. Granier et M. Kleiner. – Experimental auralization of car audio installations. *J. of the Audio Engineering Society*, vol. 44, n° 10, pp. 835–849, 1996.

[GLGM06] N. Govindaraju, K. Larsen, S. Gray et D. Manocha. – *A memory model for scientific algorithms on graphics processors*. – Rapport technique, Univ. of North Carolina at Chapel Hill, 2006.

[GLT05] Emmanuel Gallo, Guillaume Lemaitre et Nicolas Tsingos. – Prioritizing signals for selective real-time audio processing. *In : proceedings of Intl. Conf. on Auditory Display (ICAD) 2005, Limerick, Ireland*.

[GM94] Bill Gardner et Keith Martin. – *HRTF Measurements of a KEMAR Dummy-Head Microphone*. – Rapport technique n° 280, M.I.T. Media Lab Perceptual Computing, mai 1994.

[GT04] E. Gallo et N. Tsingos. – Efficient 3D audio processing with the GPU. *ACM Workshop on General Purpose Computing on Graphics Processors, Los Angeles,*, août 2004.

[HAA97] Youichi Horry, Ken-Ichi Anjyo et Kiyoshi Arai. – Tour into the picture: using a spidery mesh interface to make animation from a single image. *In : SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. pp. 225–232. – New York, NY, USA, 1997.

[Hai91] Eric Haines. – Beams O' Light: Confessions of a hacker. *Frontiers in Rendering, Course Notes, SIGGRAPH'91*, 1991.

[Han81] Robert C. Hansen (coordonnateur). – *Geometrical Theory of Diffraction*. – IEEE Press, 1981.

[HB96] Claudia M. Hendrix et Woodrow Barfield. – Presence within virtual environments as a function of visual display parameters. *Presence*, vol. 5, n° 3, pp. 274–289, 1996.

[HBd01] E. Hulsebos, E. Bourdillat et D. deVries. – Improved microphone array configurations for auralization of sound fields by wave field synthesis.

[HC95] Ellen C. Haas et John C. Casali. – Peceived urgency of and response time to multi-tone and frequency-modulated warning signals in broadband noise. *Ergonomics*, vol. 38, n° 11, pp. 2313–2326, 1995.

[HD07] J. Herre et S. Disch. – New concepts in parametric coding of spatial audio: From SAC to SAOC. *In : Proceedings of the IEEE International Conference on Multimedia and Expo*.

[Hei93] R. Heinz. – Binaural room simulation based on an image source model with addition of statistical methods to include the diffuse sound scattering of walls and to predict the reverberant tail. *Applied Acoustics*, vol. 38, pp. 145–159, 1993.

[Her99] Jens Herder. – Optimization of sound spatialization resource management through clustering. *The Journal of Three Dimensional Images, 3D-Forum Society*, vol. 13, n° 3, pp. 59–65, septembre 1999.

[HH84] P. Heckbert et P. Hanrahan. – Beam tracing polygonal objects. *Computer Graphics (SIGGRAPH 84)*, vol. 18, n° 3, pp. 119–127, juillet 1984.

[HKP+99] U. Horbach, A. Karamustafaoglu, R. Pellegrini, P. Mackensen et G. Theile. – Design and applications of a data-based auralization system for surround sound. *106th Convention of the Audio Engineering Society, preprint 4976*, 1999.

[Hob55] E. Hobson. – *The Theory of Spherical and Ellipsoidal Harmonics*. – Chelsea Pub Co., 1955.

[HSP99] Perfecto Herrera, Xavier Serra et Geoffroy Peeters. – Audio descriptors and descriptors schemes in the context of MPEG-7. *In : Procedings of International Computer Music Conference (ICMC99)*.

[HWaBS+03] W.D. Hairston, M.T. Wallace, J.W. Vaughan amd B.E. Stein, J.L. Norris et J.A. Schirillo. – Visual localization ability influences cross-modal bias. *J. Cogn. Neuroscience*, vol. 15, pp. 20–29, 2003.

[IAS] Interactive Audio Special Interest Group (IASIG), 3D Audio Working Group. http://www.iasig.org/.

[JAS83] J.-P. Jullien, A. Gilloire A. et A. Saliou. – Caractérisation d'une méthode de mesure de réponse impulsionnelle en acoustique des salles. pp. 217–220.

[JCW97] J.-M. Jot, L. Cerveau et O. Warusfel. – Analysis and synthesis of room reverberation based on a time-frequency model. *In : AES 103rd convention preprint*. AES. – preprint #4629.

[Jeh05] T. Jehan. – *Creating Music by Listening*. – Thèse de doctorat, M.I.T., juin 2005.

[JJ00] A. Jost et J.-M. Jot. – Transaural 3-d audio with user-controlled calibration. *Proceedings of COST-G6 Conference on Digital Audio Effects, DAFX2000, Verona, Italy*, december 2000.

[JLP99] J.-M. Jot, V. Larcher et J.-M. Pernaux. – A comparative study of 3D audio encoding and rendering techniques. *Proceedings of the AES 16th international conference, Spatial sound reproduction, Rovaniemi, Finland*, april 1999.

[JM82] G.M. Jebsen et H. Medwin. – On the failure of the Kirchhoff assumption in backscatter. *J. Acous. Soc. of America*, no5, pp. 1607–1611, Nov 1982.

[JMT03] C. Joslin et N. Magnenat-Thalmann. – Significant fact retreival for real-time 3D sound rendering in complex virtual environments. *Proc. of VRTST 2003*, October 2003.

[Jon71] C. B. Jones. – A new approach to the 'hidden line' problem. *Computer Journal*, vol. 14, n° 3, pp. 232–237, août 1971.

[Jot92a] Jean-Marc Jot. – *Etude et réalisation d'un spatialisateur de sons par modèles physique et perceptifs*. – Thèse de doctorat, Ecole Normale Supérieure des Télécommunications, Paris, 1992.

[Jot92b] J.M. Jot. – An analysis/synthesis approach to real-time artificial reverberation. *Proc. of ICASSP*, 1992.

[Jot97] J.-M. Jot. – Efficient models for reverberation and distance rendering in computer music and virtual audio reality. Proceedings of Intl. Computer Music Conf.

[Jot99] Jean-Marc Jot. – Real-time spatial processing of sounds for music, multimedia and interactive human-computer interfaces. *Multimedia Systems*, vol. 7, n° 1, pp. 55–69, 1999.

[JRY00] Alexander Jourjine, Scott Rickard et Ozgur Yilmaz. – Blind separation of disjoint orthogonal signals: Demixing n sources from 2 mixtures. *In : IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'00), Istanbul, Turkey*.

[JW95] J.M. Jot et O. Warusfel. – Le spatialisateur. *Proc. of Rencontres Musicales Pluridisciplinaires, Le Son & L'Espace*, pp. 103–108, Lyon, mars 1995.

[JWL98] J.-M. Jot, S. Wardle et V. Larcher. – Approaches to binaural synthesis. *Proc. of the 105th AES Convention, preprint 4861*, 1998.

[KAG+02] Evelyn Kurniawati, Javed Absar, Sapna George, Chiew Tong Lau et Benjamin Premkumar. – The significance of tonality index and nonlinear psychoacoustics models for masking threshold estimation. *In : Proceedings of the International Conference on Virtual, Synthetic and Entertainment Audio AES22*.

[Kat01a] B. Katz. – Boundary element method calculation of individual head-related transfer function. part i: Rigid model calculation. *J. of the Acoustical Society of America*, vol. 110, n° 5, pp. 2440–2448, novembre 2001.

[Kat01b] B. Katz. – Boundary element method calculation of individual head-related transfer function. part ii: Impedance effects and comparison to real measurements. *J. of the Acoustical Society of America*, vol. 110, n° 5, pp. 2449–2455, novembre 2001.

[Kaw81] T. Kawai. – Sound diffraction by a many sided barrier or pillar. *J. of Sound and Vibration*, vol. 79, n° 2, pp. 229–242, 1981.

[KDS93] M. Kleiner, B.I. Dalenbäk et P. Svensson. – Auralization - an overview. *J. of the Audio Engineering Society*, vol. 41, n° 11, pp. 861–875, novembre 1993.

[Kel62] J.B. Keller. – Geometrical theory of diffraction. *J. of the Optical Society of America*, vol. 52, n° 2, pp. 116–130, 1962.

[kem] KEMAR HRTF database. Available from http://sound.media.mit.edu/KEMAR.html.

[KG83] G.F. Kuhn et R.M. Guernsey. – Sound pressure distribution around the human head and torso. *J. of the Acoustical Society of America*, vol. 73, pp. 95–105, 1983.

[KGW+99] S.C. Kim, B. Guarino, T. Willis, V. Erceg, S. Fortune, R. Valenzuela, L. Thomas, J. Ling et J. Moore. – Radio propagation measurements and prediction using three-dimensional ray tracing in urban environments at 908 MHz and 1.9 GHz. *IEEE Trans. on Vehicular Technology*, vol. 48, pp. 931–946, 1999.

[KKF93] U.R. Kristiansen, A. Krokstad et T. Follestad. – Extending the image method to higher-order reflections. *J. Applied Acoustics*, vol. 38, n° 2–4, pp. 195–206, 1993.

[KNPC99] Y. Kahana, P.A. Nelson, M. Petyt et S. Choi. – Numerical modelling of the transfer functions of a dummy-head and of the external ear. *Proceedings of the AES 16th international conference, Spatial sound reproduction, Rovaniemi, Finland*, pp. 330–345, april 1999.

[KOK93] M. Kleiner, R. Orlowski et J. Kirszenstein. – A comparison between results from a physical scale model and a computer image source model for architectural acoustics. *Applied Acoustics*, vol. 38, pp. 245–265, 1993.

[KP74] Robert G. Kouyoumjian et Prabhakar H. Pathak. – A uniform geometrical theory of diffraction for an edge in a perfectly conducting surface. *Proc. of IEEE*, vol. 62, pp. 1448–1461, novembre 1974.

[KPLL05] C. Kayser, C. Petkov, M. Lippert et N.K. Logothetis. – Mechanisms for allocating auditory attention: An auditory saliency map. *Current Biology*, vol. 15, pp. 1943–1947, novembre 2005.

[KT02] M.C. Kelly et A.I. Tew. – The continuity illusion in virtual auditory space. *proc. of the 112th AES Conv., Munich, Germany*, mai 2002.

[KUG93] P. Kreuzgruber, P. Unterberger et R. Gahleitner. – A ray splitting model for indoor radio propagation associated with complex geometries. *Proceedings of the 1993 43rd IEEE Vehicular Technology Conference*, pp. 227–230, 1993.

[Kut04] Heinrich Kuttruff. – *Room Acoustics (4th edition)*. – Taylor & Francis, 2004.

[Lab] Sound Lab.
http://human-factors.arc.nasa.gov/SLAB/.

[Lar01] Vronique Larcher. – *Techniques de spatialisation des sons pour la ralit virtuelle*. – Thèse de doctorat, Universit Pierre et Marie Curie, IRCAM, Paris, 2001.

[LBM03] A. Laborie, R. Bruno et S. Montoya. – A new comprehensive approach of surround sound recording. *Proc. 114th convention of the Audio Engineering Society, preprint 5717*, 2003.

[LBM04] A. Laborie, R. Bruno et S. Montoya. – High spatial resolution multi-channel recording. *Proc. 116th convention of the Audio Engineering Society, preprint 6116*, 2004.

[LCM07] C. Lauterbach, A. Chandak et D. Manocha. – Interactive sound rendering in complex and dynamic scenes using frustum tracing. *Transactions on Visualization and Computer Graphics*, vol. 13, n° 6, pp. 1672–1679, Nov.-Dec. 2007.

[Lee98] Martin J. Leese. – Ambisonic surround sound FAQ (version 2.8), 1998. http://members.tripod.com/martin_leese/Ambisonic/.

[LEG01] Jrg Lewald, Walter H. Ehrenstein et Rainer Guski. – Spatio-temporal constraints for auditory-visual integration. *Beh. Brain Research*, vol. 121, n° 1-2, pp. 69–79, 2001.

[Leh93] H. Lehnert. – Systematic errors of the ray-tracing algorithm. *Applied Acoustics*, vol. 38, 1993.

[Lew93] T. Lewers. – A combined beam tracing and radiant exchange computer model of room acoustics. *Applied Acoustics*, vol. 38, 1993.

[LG95] David Luebke et Chris Georges. – Portals and mirrors: Simple, fast evaluation of potentially visible sets. *In : 1995 Symposium on Interactive 3D Graphics*, éd. par Pat Hanrahan et Jim Winget. ACM SIGGRAPH, pp. 105–106.

[LGST00] T. Lokki, M. Gröhn, L. Savioja et T. Takala. – A case study of auditory navigation in virtual acoustic environments. *Proceedings of Intl. Conf. on Auditory Display*, vol. (ICAD2000), 2000.

[LH96] Marc Levoy et Pat Hanrahan. – Light field rendering. *In : SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. pp. 31–42. – New York, NY, USA, 1996.

[LHS01] T. Lokki, T. Hiipalla et L. Savioja. – A framework for evaluating virtual acoustic environments. *AES 110th convention, Berlin, preprint 5317*, 2001.

[lis] LISTEN HRTF database.
Available from http://recherche.ircam.fr/equipes/salles/listen.

[LM01] Mathieu Lagrange et Sylvain Marchand. – Real-time additive synthesis of sound by taking advantage of psychoacoustics. *In : Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-01), Limerick, Ireland, December 6-8*.

[LNV+04] T. Lokki, H. Nironen, S. Vesa, L. Savioja, A. Härmä et M. Karjalainen. – Application scenarios of wearable and mobile augmented reality audio. *AES 116th convention, Berlin, Germany*, mai 2004.

[Log00] Beth Logan. – Mel frequency cepstral coefficients for music modeling. *In : Proceedings of the International Symposium on Music Information Retrieval (Music IR 2000), Plymouth Massaschussetts october 23-25 2000*.

[LRC+02] D. Luebke, M. Reddy, J. Cohen, A. Varshney, B. Watson et R. Huebner. – *Level of Detail for 3D Graphics*. – Morgan Kaufmann, Computer Graphics and Geometric Modeling series, 2002.

[LS97] J. Lavergnat et M. Sylvain. – *Propagation des ondes radioelectriques - introduction*. – Masson ed., 1997.

[LS00] M. S. Lewicki et T. J. Sejnowski. – Learning overcomplete representations. *Neural Computation*, vol. 12, n° 2, pp. 337–365, 2000.

[LSVA07] Tobias Lentz, Dirk Schröder, Michael Vorländer et Ingo Assenmacher. – Virtual reality system with integrated sound field simulation and reproduction. *EURASIP Journal on Advances in Signal Processing*, vol. 2007, pp. Article ID 70540, 19 pages, 2007. – doi:10.1155/2007/70540.

[LVK02] P. Larsson, D. Västfjäll et M. Kleiner. – Better presence and performance in virtual environments by improved binaural sound rendering. *proceedings of the AES 22nd Intl. Conf. on virtual, synthetic and entertainment audio, Espoo, Finland*, pp. 31–38, juin 2002.

[LWZ04] L. Lu, L. Wenyin et H.-J. Zhang. – Audio textures: Theory and applications. *IEEE Transactions on Speech and Audio Processing*, vol. 12, n° 2, pp. 156–167, 2004.

[LYLG03] Wen-Chieh Lee, Chung-Han Yang, Chi-Min Liu et Juin-In Guo. – Perceptual convolution for reverberation. *In : In Proceeding of 115th AES Convention, Los Angeles*.

[MAB+03] M.Rath, F. Avanzini, N. Bernardini, G. Borin, F. Fontana, L. Ottaviani et D. Rochesso. – An introductory catalog of computer-synthesized contact sounds, in real-time. *Proc. of the XIV Colloquium on Musical Informatics, Firenze, Italy*, juillet 2003.

[Mal01] D.G. Malham. – Spherical harmonic coding of sound objects - the ambisonic 'O' format. *Proc. of the 19th AES Conference, Surround Sound, Techniques, Technology and Perception, Schloss Elmau, Germany*, juin 2001.

[MBT+07] Thomas Moeck, Nicolas Bonneel, Nicolas Tsingos, George Drettakis, Isabelle Viaud-Delmon et David Aloza. – Progressive perceptual audio rendering of complex scenes. *In : Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*. ACM SIGGRAPH.

[ME04a] J. Meyer et G. Elko. – Spherical microphone arrays for 3d sound recording. *chap. 2 in Audio Signal Processing for next-generation multimedia communication systems, Eds. Yiteng (Arden) Huang and Jacob Benesty, Bosten, Kluwer Academic Publisher*, 2004.

[ME04b] J. Meyer et G.W. Elko. – *Spherical microphone arrays for 3D sound recording, chap. 2 in Audio Signal Processing for next-generation multimedia communication systems, Eds. Yiteng (Arden) Huang and Jacob Benesty, Bosten, Kluwer Academic Publisher*. – 2004.

[Men02] D. Menzies. – W-Panning and O-format, tools for object spatialization. *Proceedings of Intl. Conf. on Auditory Display*, 2002.

[Mer02] J. Merimaa. – Applications of a 3D microphone array. *112th AES convention, preprint 5501*, 2002.

[MF00] P. Min et T. Funkhouser. – Priority-driven acoustic modeling for virtual environments. *Computer Graphics Forum, Proceedings of EUROGRAPHICS 2000*, vol. 19, n° 3, juillet 2000.

[MGB97] B. C. J. Moore, B. Glasberg et T. Baer. – A model for the prediction of thresholds, loudness and partial loudness. *J. of the Audio Engineering Society*, vol. 45, n° 4, pp. 224–240, 1997. – Software available at http://hearing.psychol.cam.ac.uk/Demos/demos.html.

[MHJS95] H. Møller, D. Hammershøi, C.B. Jensen et M.F. Sørensen. – Transfer characteristics of headphones measured on human ears. *J. of the Audio Engineering Society*, vol. 43, n° 4, pp. 203–217, avril 1995.

[Mil01] M. Miller. – Slab: a software-based real-time virtual acoustic environment rendering system. *Proceedings of Intl. Conf. on Auditory Display*, 2001.

[Mit98] Joseph S. B. Mitchell. – Geometric shortest paths and network optimization. *In : Handbook of Computational Geometry*, éd. par Jörg-Rüdiger Sack et Jorge Urrutia. – Amsterdam, Elsevier Science Publishers B.V. North-Holland, 1998.

[MM95] D.G. Malham et A. Myatt. – 3D sound spatialization using ambisonic techniques. *Computer Music Journal*, vol. 19, n° 4, pp. 58–70, 1995.

[MOD96] M. Monks, B.M. Oh et J. Dorsey. – Acoustic simulation and visualisation using a new unified beam tracing and image source approach. *Proc. Audio Engineering Society Convention*, 1996.

[Møl89] Henrik Møller. – Reproduction of artificial-head recordings through loudspeakers. *J. of the Audio Engineering Society*, vol. 37, n° 1/2, pp. 30–33, jan/feb 1989.

[Møl92] Henrik Møller. – Fundamentals of binaural technology. *Applied Acoustics*, vol. 36, pp. 171–218, 1992.

[Moo79] J.A. Moorer. – About this reverberation business. *Computer Music Journal*, vol. 23, n° 2, 1979.

[Moo97] Brian C.J. Moore. – *An introduction to the psychology of hearing*. – Academic Press, 4th edition, 1997.

[MPM90] D.A. McNamara, C.W.I. Pistorius et J.A.G. Malherbe. – *Introduction to the Uniform Geometrical Theory of Diffraction*. – Artech House, 1990.

[MSHJ95] H. Møller, M.F. Sørensen, D. Hammershøi et C.B. Jensen. – Head-related transfer functions of human subjects. *J. of the Audio Engineering Society*, vol. 43, n° 5, pp. 300–321, mai 1995.

[MvMV93] J. Martin, D. van Maercke et J.P. Vian. – Binaural simulation of concert halls : A new approach for the binaural reverberation process. *J. of the Acoustical Society of America*, vol. 94, pp. 3255–3263, décembre 1993.

[MW02] M. Miller et E. Wenzel. – Recent developments in slab: A software-based system for interactive spatial sound synthesis. *Proceedings of Intl. Conf. on Auditory Display*, 2002.

[MW04] Wolfgang Straßer Michael Wand. – Multi-resolution sound rendering. *In : Symp. Point-Based Graphics*.

[MZ93] S. Mallat et Z. Zhang. – Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, vol. 41, n° 12, pp. 3397–3415, 1993.

[Nay93] J.M. Naylor. – Odeon - another hybrid room acoustical model. *Applied Acoustics*, vol. 38, n° 1, pp. 131–143, 1993.

[NNK93] G.V. Norton, J.C. Novarini et R.S. Keiffer. – An evaluation of the Kirchhoff approximation in predicting the axial impulse response of hard and soft disks. *J. Acous. Soc. of America*, no6, pp. 3094–3056, June 1993.

[NTLS07] Arnault Nagle, Nicolas Tsingos, Guillaume Lemaitre et Aurelien Sollaud. – On-the-fly auditory masking for scalable voip bridges. *In : AES 30th International Conference on Intelligent Audio Environments*.

[OCE01] James F. O'Brien, Perry R. Cook et Georg Essl. – Synthesizing sounds from physically based motion. *ACM Computer Graphics, SIGGRAPH'01 Proceedings*, pp. 545–552, août 2001.

[OHL+08] John D. Owens, Mike Houston, David Luebke, Simon Green, John E. Stone et James C. Phillips. – GPU computing. *Proceedings of the IEEE*, vol. 96, n° 5, mai 2008.

[OLG+05] John D. Owens, David Luebke, Naga Govindaraju, Mark Harris, Jens Krüger, Aaron E. Lefohn et Tim Purcell. – A survey of general-purpose computation on graphics hardware. *In : Eurographics 2005, State of the Art Reports*, pp. 21–51.

[OPE00] OpenAL: an open source 3D sound library, 2000. http://www.openal.org.

[OSG02] J.F. O'Brien, C. Shen et C.M. Gatchalian. – Synthesizing sounds from rigid-body simulations. *Proc. of the ACM SIGGRAPH Symposium on Computer Animation, San Antonio, Texas*, pp. 175–182, juillet 2002.

[PB04a] F. Pachet et J-P. Briot (coordonnateur). – *Informatique Musicale - du signal au signe musical*. – Hermès science, 2004.

[PB04b] J. R. Parker et B. Behm. – Generating audio textures by example,. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'04)*, 2004.

[PB04c] G. Potard et I. Burnett. – Decorrelation techniques for the rendering of apparent source width in 3D audio displays. *Proc. of 7th Intl. Conf. on Digital Audio Effects (DAFX'04), Naples, Italy*, octobre 2004.

[PC03] J.R. Parker et S. Chan. – Sound synthesis for the web, games, and virtual reality. *International Conference on Computer Graphics and Interactive Techniques*, 2003.

[Pee04] Geoffroy Peeters. – *A large set of audio features for sound description (similarity and classification) in the CUIDADO project*. – Cuidado projet report, Institut de Recherche et de Coordination Acoustique Musique (IRCAM), 2004.

[Pel99] R.S. Pellegrini. – Comparison of data and model-based simulation algorithms for auditory virtual environments. *106th Convention of the Audio Engineering Society, preprint 4953*, 1999.

[Pel01a] R. Pellegrini. – Quality assessment of auditory virtual environments. *Proceedings of Intl. Conf. on Auditory Display*, vol. (ICAD2001), 2001.

[Pel01b] R.S. Pellegrini. – *A virtual Listening Room as an application of auditory virtual Environment*. – Thèse de doctorat, PhD dissertation, Ruhr-Universität, Bochum, 2001.

[PF06] V. Pulkki et C. Faller. – Directional audio coding: Filterbank and stft-based design. *In 120th AES Convention, Paris, France, Preprint 6658.*, May 20-23 2006.

[Pie84] A.D. Pierce. – *Acoustics. An introduction to its physical principles and applications*. – 3rd edition, American Institute of Physics, 1984.

[Pom99] F.J. Pompei. – The use of airborne ultrasonics for generating audible sound beams. *J. of the Audio Engineering Society*, vol. 47, n° 9, pp. 726–731, 1999.

[PS00] E. M. Painter et A. S. Spanias. – Perceptual coding of digital audio. *Proceedings of the IEEE*, vol. 88, n° 4, avril 2000.

[Pul97] Ville Pulkki. – Virtual sound source positioning using vector base amplitude panning. *J. of the Audio Engineering Society*, vol. 45, n° 6, pp. 456–466, juin 1997.

[Pul06] V. Pulkki. – Directional audio coding in spatial sound reproduction and stereo upmixing. *Proc. of the AES 28th Int. Conf, Pitea, Sweden*, June 2006.

[Ran01] Ramapriya Rangachar. – *Analysis and Improvement of the MPEG-1 Audio Layer III Algorithm at Low Bit-Rates*. – Master of science thesis, Arizona State University, décembre 2001.

[RBF03] D. Rocchesso, R. Bresin et M. Frenström. – Sounding objects. *IEEE Multimedia*, vol. 10, n° 2, pp. 42–52, avril 2003.

[RNFR96] A. Rajkumar, B.F. Naylor, F. Feisullin et L. Rogers. – Predicting RF coverage in large environments using ray-beam tracing and partitioning tree represented geometry. *Wireless Networks*, vol. 2, n° 2, pp. 143–154, 1996.

[Roa96] C. Roads. – *The computer music tutorial*. – MIT Press, 1996.

[RR02] Richard Radke et Scott Rickard. – Audio interpolation. *In : the Audio Engineering Society 22nd International Conference on Virtual, Synthetic and Entertainment Audio (AES'22), Espoo, Finland*, pp. 51–57.

[SAP95] Nicolas Saint-Arnaud et Kris Popat. – Analysis and synthesis of sound textures. *International Joint Conference on Neural Networks (IJCAI'95),Montreal*, pp. 125–131, 1995.

[Sch62] M.R. Schroeder. – Natural sounding artificial reverberation. *J. of the Audio Engineering Society*, vol. 10, n° 3, pp. 219–223, 1962.

[SE98] R. Streicher et F.A. Everest (coordonnateur). – *The new stereo soundbook, 2nd edition*. – Audio Engineering Associate, Pasadena (CA), USA, 1998.

[Sen01] Sensaura. – ZoomFX, MacroFX, Sensaura©, 2001.
http://www.sensaura.co.uk.

[SF03] B.U. Seeber et H. Fastl. – Subjective selection of non-individual head-related transfer functions. *Proceedings of Intl. Conf. on Auditory Display*, Boston, USA, July 2003 2003.

[SFV99] U. P. Svensson, R. I. Fred et J. Vanderkooy. – Analytic secondary source model of edge diffraction impulse responses. *J. Acoust. Soc. Am.*, vol. 106, pp. 2331–2344, 1999.

[SHLV99] L. Savioja, J. Huopaniemi, T. Lokki et R. Väänänen. – Creating interactive virtual acoustic environments. *J. of the Audio Engineering Society*, vol. 47, n° 9, pp. 675–705, septembre 1999.

[Sib01] A. Sibbald. – MacroFX algorithm. white paper. *http://www.sensaura.co.uk/whitepapers*, 2001.

[Sil05] S. Siltanen. – Geometry reduction in room acoustics modeling. *Master Thesis, Helsinki University Of Technology, Department of Computer Science Telecommunications Software and Multimedia Laboratory*, September 2005.

[SK95] Uwe Stevenson et Ulf Kristiansen. – Pyramidal beam tracing and time dependent radiosity. *Proc. of 15th International Congress on Acoustics*, pp. 657–660, Trondheim, Norway, juin 1995.

[SN81] Y. Sakurai et K. Nagata. – Sound reflections of a rigid plane panel and of the "live-end" composed by those panels. *J. Acous. Soc. of Japan*, no1, pp. 5–14, Jan 1981.

[Sou] Soundfield. http://www.soundfield.com.

[Sou04] SoundBlaster. – Creative Labs Soundblaster©, 2004. http://www.soundblaster.com.

[SP82] J. Stautner et M. Puckette. – Designing multi-channel reverbertors. *Computer Music Journal*, vol. 6, n° 1, 1982.

[SP94] François X. Sillion et C. Puech. – *Radiosity and Global Illumination*. – Morgan Kaufmann Publishers inc., 1994.

[spa] Spatialisateur de l'IRCAM. http://recherche.ircam.fr/equipes/salles/projets/Spatialisateur.html.

[SRT94] L. Savioja, T. Rinne et T. Takala. – Simulation of room acoustics with a 3D finite difference mesh. *Proceedings of Intl. Computer Music Conf.*, vol. (ICMC94), pp. 463–466, septembre 1994.

[ste89] Stereophonic Techniques - *An anthology of reprinted articles on stereophonic techniques*. – Audio Engineering Society, 1989.

[Stra] R. Streicher. – The decca tree. http://mixonline.com/recording/applications/audio_decca_tree/.

[Strb] R. Streicher. – The decca tree – it's not just for stereo anymore. http://www.wesdooley.com/pdf/Surround_Sound_Decca_Tree-urtext.pdf.

[SV01] L. Savioja et V. Välimäki. – Interpolated 3D digital waveguide mesh with frequency warping. *Proceedings of Intl. Conf. on Acoustics, Speech and Signal Processing*, vol. Salt Lake City, mai 2001.

[TDLD07] Nicolas Tsingos, Carsten Dachsbacher, Sylvain Lefebvre et Matteo Dellepiane. – Instant sound scattering. *In : Rendering Techniques (Proceedings of the Eurographics Symposium on Rendering)*.

[tdt] Tucker-Davis Technologies. http://www.tdt.com/.

[Tel92] Seth Teller. – Computing the antiumbra cast by an area light source. *Computer Graphics (SIGGRAPH 92)*, vol. 26, n° 2, pp. 139–148, 1992.

[TEP04] Abdellatif B. Touimi, Marc Emerit et Jean-Marie Pernaux. – Efficient method for multiple compressed audio streams spatialization. *In : In Proceeding of ACM 3rd Intl. Conf. on Mobile and Ubiquitous multimedia*.

[TFNC01] N. Tsingos, T. Funkhouser, A. Ngan et I. Carlbom. – Modeling acoustics in virtual environments using the uniform theory of diffraction. *ACM Computer Graphics, SIGGRAPH'01 Proceedings*, pp. 545–552, août 2001.

[TG97] Nicolas Tsingos et Jean-Dominique Gascuel. – Soundtracks for computer animation: sound rendering in dynamic environments with occlusions. *Proceedings of Graphics Interface'97*, pp. 9–16, mai 1997.

[TG98] Nicolas Tsingos et Jean-Dominique Gascuel. – Fast rendering of sound occlusion and diffraction effects for virtual acoustic environments. *Proc. 104th Audio Engineering Society Convention, preprint 4699*, preprint n° 4699 (P4-7), Amsterdam, Netherlands, mai 1998.

[TGD04] N. Tsingos, E. Gallo et G. Drettakis. – Perceptual audio rendering of complex virtual environments. *ACM Transactions on Graphics, Proceedings of SIGGRAPH 2004*, août 2004.

[Tho87] E.I. Thorsos. – The validity of the Kirchhoff approximation for rough surface scattering using a gaussian roughness spectrum. *J. Acous. Soc. of America*, no1, pp. 78–92, Jan 1987.

[TNKH98] Takashi Takeushi, P.A. Nelson, O. Kirkeby et H. Hamanda. – Influence of individual head related transfer function on the performance of virtual acoustic imaging systems. *Proc. 104th Audio Engineering Society Convention*, preprint n° 4700 (P4-3), Amsterdam, Netherlands, mai 1998.

[Tou00] Abdellatif B. Touimi. – A generic framework for filtering in subband domain. *In : In Proceeding of IEEE 9th Workshop on Digital Signal Processing, Hunt, Texas, USA*.

[TRI] TRINNOV Audio. http://www.trinnov.com.

[Tsi05a] N. Tsingos. – Scalable perceptual mixing and filtering of audio signals using an augmented spectral representation. *Proc. of 8th Intl. Conf. on Digital Audio Effects (DAFX'05), Madrid, Spain*, septembre 2005.

[Tsi05b] Nicolas Tsingos. – Scalable perceptual mixing and filtering of audio signals using an augmented spectral representation. *In : Proceedings of the International Conference on Digital Audio Effects*. – Madrid, Spain.

[vdDKP01] Kees van den Doel, Paul G. Kry et Dinesh K. Pai. – Foleyautomatic: Physically based sound effects for interactive simulation and animation. *ACM Computer Graphics, SIGGRAPH'01 Proceedings*, pp. 545–552, août 2001.

[vdDKP04] Kees van den Doel, Dave Knott et Dinesh K. Pai. – Interactive simulation of complex audio-visual scenes. *Presence: Teleoperators and Virtual Environments*, vol. 13, n° 1, 2004.

[vdDPA+02] K. van den Doel, D. K. Pai, T. Adam, L. Kortchmar et K. Pichora-Fuller. – Measurements of perceptual quality of contact sound models. *In : Proceedings of the International Conference on Auditory Display (ICAD 2002), Kyoto, Japan*, pp. 345–349.

[VJGW00] V.Larcher, J.M. Jot, G. Guyard et O. Warusfel. – Study and comparison of efficient methods for 3d audio spatialization based on linear decomposition of HRTF data. *Proc. 108th Audio Engineering Society Convention*, preprint n° 5097 (E1), Paris, 2000.

[vMM93] D. van Maercke et J. Martin. – The prediction of echograms and impulse responses within the Epidaure software. *Applied Acoustics*, vol. 38, n° 1, pp. 93–114, 1993.

[VRR+03] E. Vincent, X. Rodet, A. Röbel, C. Févotte, E. Le Carpentier, R. Gribonval, L. Benaroya et F. Bimbot. – A tentative typologie of audio source separation tasks. *Proc. of the 4th Intl. Symposium on Independent Component Analysis and Blind Signal Separation (ICA2003), Nara, Japan*, avril 2003.

[Wat90] Mark Watt. – Light-water interaction using backward beam tracing. *ACM Computer Graphics, SIGGRAPH '90 Proceedings*, pp. 377–385, août 1990.

[Wen01] E. Wenzel. – Effect of increasing system latency on localization of virtual sounds with short and long duration. *Proceeding of ICAD 2001, Espoo, Finland*, august 2001.

[WK89] F. Wightman et D.J. Kistler. – Headphone simulation of free-field listening (part 1 and 2). *J. of the Acoustical Society of America*, vol. 85, n° 2, pp. 858–878, février 1989.

[WK05] F. Wightman et D. Kistler. – Measurement and validation of human HRTFs for use in hearing research. *Acustica, Acta-Acustica*, vol. 91, pp. 429–439, 2005.

[WM04] O. Warusfel et N. Misdariis. – Sound source radiation synthesis : from stage performance to domestic rendering.

[WRR04] L.M. Wang, J. Rathsam et S.R. Ryherd. – Interactions of model detail level and scattering coefficients in room acoustic computer simulation. *Intl. Symp. on Room Acoustics, a satelite symposium of ICA, Kyoto, Japan*, 2004.

[ZB94] Udo Zölzer et Thomas Bolze. – Interpolation algorithms: theory and applications. *Proc. 97th Audio Engineering Society Convention, preprint 3898*, preprint n° 3898 (K-3), San Francisco, USA, novembre 1994.

[Zöl02] Udo Zölzer (coordonnateur). – *DAFX - Digital Audio Effects*. – Wiley, 2002.

List of Figures