

Recherche des intersections rayon - surfaces par classement préférentiel dans un logiciel d'acoustique des salles

Stéphane Lesoinne¹, Jean-Jacques Embrechts²

¹Intelsig, Institut Montéfiore, B28, Université de Liège, B-4000 Liège, Belgique s.lesoinne@ulg.ac.be

²Intelsig, Institut Montéfiore, B28, Université de Liège, B-4000 Liège, Belgique jjembrechts@ulg.ac.be

Dans un algorithme classique de tir de rayons sonores, la recherche de l'intersection entre un segment de rayon sonore et la prochaine surface est effectuée en testant l'intersection de ce rayon avec toutes les surfaces présentes. Cependant, dans une salle quelconque, la probabilité d'intersection d'un rayon avec le modèle géométrique n'est pas uniformément répartie entre les surfaces. Pour chaque surface « i » origine d'un segment de rayon, l'idée est d'obtenir la liste des surfaces « j » triée par ordre de probabilité décroissante d'intersection. Cette probabilité d'intersection peut-être estimée à partir d'un petit sous-ensemble du nombre total de rayons sonores tirés par la source acoustique. Lors d'une recherche, la liste des surfaces triées peut être parcourue selon différentes stratégies : - Assigner la première surface trouvée dans la liste comme surface destination, au prix d'une imprécision sur les trajets des rayons. En fonction du soin apporté à la modélisation de la salle, cette solution peut entraîner la perte de nombreux rayons. - Interrompre la recherche à la première intersection trouvée et chercher s'il existe une intersection avec les surfaces intermédiaires, surfaces situées entre la surface origine du segment et la surface destination trouvée. La liste des surfaces intermédiaires est obtenue avant le tir de rayons au moyen d'enveloppes convexes. Cette modification permet de ne perdre aucun rayon et de respecter le trajet initial des rayons.

1 Introduction

Le tir de rayons est une des méthodes les plus populaires pour calculer des champs sonores et des réponses impulsionnelles de salle. Il y a plusieurs façons d'implémenter les algorithmes de tir de rayons [1, 2], la plupart d'entre elles tentant d'obtenir les résultats les plus précis dans le temps de calcul le plus court possible.

Parmi toutes ces méthodes, le tir de rayons aléatoires consiste à simuler le trajet d'un grand nombre de rayons sonores dans le volume d'une salle, ces rayons étant émis par la source dans des directions aléatoires. Pour calculer les niveaux de pression sonore (SPL) à une position donnée dans la salle, un récepteur (plan, sphérique, ...) collecte l'énergie des rayons passant à proximité et enregistre le temps d'arrivée. Les trajets de ces rayons sont définis par leur réflexion sur les surfaces constituant la salle et sont calculés jusqu'à extinction du rayon (énergie transportée trop faible ou nombre de réflexions supérieur à une borne maximale). La majeure partie du temps de calcul est consacrée à la recherche d'intersections entre les rayons et les surfaces, d'autant plus que le nombre de surfaces, de rayons et de réflexions est élevé.

On propose une méthode de réduction du nombre de surfaces planes traitées lors de la recherche d'intersection. Elle est basée sur une classification des surfaces suivant une pseudo probabilité d'intersection, couplée à une détection des surfaces potentiellement obstructives.

2 Recherche de l'intersection des rayons avec les surfaces dans l'algorithme traditionnel

Lorsqu'un rayon sonore est émis d'une source, son parcours est régi par les réflexions¹ sur les surfaces² (obstacles) rencontrées. Lorsqu'un segment de rayon rencontre une surface, le point d'intersection est calculé. La puissance du rayon et sa direction sont mises à jour tandis que le point d'intersection devient l'origine du nouveau segment de rayon.

Pour chaque réflexion d'un rayon sonore, il est nécessaire de calculer la surface destination, c'est à dire, la surface pour laquelle l'intersection détermine le segment de rayon le plus court. Pour ce faire, on balaie l'ensemble des surfaces et cette recherche se décompose en deux temps :

1. Chaque surface plane est incluse dans un plan de support. On calcule la distance D entre l'origine du segment de rayon et le plan de support de la surface testée.
2. Si aucune intersection avec une surface n'a encore été trouvée, on fixe « distance_ref » à une valeur arbitrairement grande (idéalement infinie). Si la distance D est positive et est inférieure à « distance_ref », on teste si le point d'intersection appartient à la surface. Si oui, la surface est marquée comme surface destination potentielle et « distance_ref » prend la valeur D . Sinon, la surface est ignorée.

¹ Prise en compte des réflexions spéculaires et diffuses [5]

² Toutes les surfaces considérées sont planes.

Lorsque toutes les surfaces ont été parcourues, on connaît la surface destination et le rayon peut être réfléchi.

2.1 Utilisation des capacités SIMD

Une première accélération de la recherche d'intersection est obtenue par l'utilisation des capacités SIMD (Single Input Multiple Data) des processeurs actuels. En effet, ces processeurs possèdent des unités de calcul capables d'effectuer certaines instructions simultanément sur plusieurs données du même type.

L'algorithme de calcul des distances est modifié pour effectuer d'un coup le calcul des distances d'intersection de toutes les surfaces. Les surfaces sont traitées par groupes de taille K , où K est le nombre de « float » traités simultanément par les instructions SIMD ($K = 4$ sur les processeurs utilisant les jeux d'instruction SSE2, ... [3]).

Le nombre de branchements est limité afin de réduire les divergences dans le flux de données fournit au processeur. Le gain apporté par cette modification sera examiné au point 5

Simulations)5 Simulations.

3 Recherche des surfaces occlusives

3.1 Concept

Dans la méthode de recherche présentée au point 2, le parcours de la liste complète des surfaces est obligatoire puisqu'on cherche l'intersection avec la surface la plus proche du point d'origine du segment du rayon sonore.

Si on acceptait de ne pas calculer le chemin « exact » pour tous les rayons, on pourrait classer les surfaces par leur probabilité d'intersection avec le rayon et terminer la recherche dès qu'une intersection est trouvée, ce qui permettrait de ne pas parcourir toute la liste des surfaces à chaque réflexion au risque d'introduire des erreurs dans le trajet du rayon.

Si cette modification est mise en œuvre à partir d'un temps de parcours du rayon suffisamment long (champ sonore s'approchant du champ diffus), son influence sur la précision des résultats de la simulation peut être acceptable.

Cependant, un problème apparaît si les surfaces débordent du volume de la salle. Les rayons peuvent accéder à des parties inaccessibles du volume suite aux erreurs de trajet et se perdre (Figure 1). Plus il y a de rayons qui se perdent et plus cela se passe tôt, moins il y aura de rayons contribuant aux récepteurs. Pour résoudre ce problème en conservant l'algorithme tel quel, il faudrait imposer des contraintes sur le design de la salle et en particulier éviter les surfaces débordantes

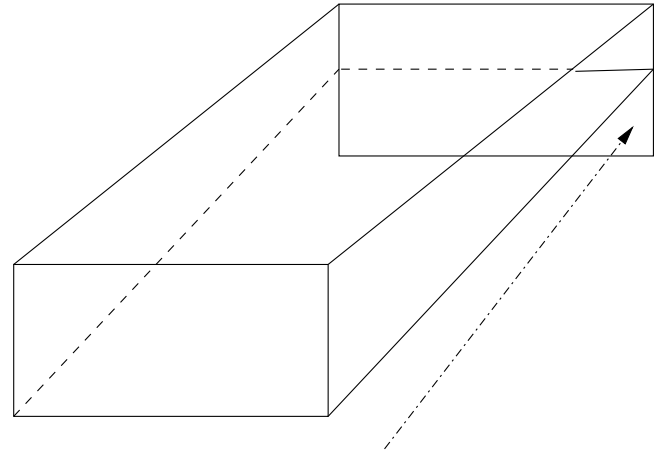


Figure 1 : Surface débordante. La surface indiquée par une flèche peut être considérée comme prochaine surface

Une autre solution, qui permet d'obtenir le chemin « exact » des rayons, utilise une connaissance rudimentaire de la topographie de la salle. Elle nécessite, pour chaque couple de surfaces non coplanaires A et B , la connaissance de la liste des surfaces visibles³ par un rayon quelconque voyageant entre A et B .

Dès que ces listes sont connues, dès qu'une intersection « temporaire » avec une surface S_2 est trouvée avec un segment de rayon originaire d'une surface S_1 , on arrête le parcours de la liste des surfaces et on continue la recherche d'une intersection plus proche sur la liste des surfaces occlusives entre S_1 et S_2 .

L'algorithme adapté avec les surfaces occlusives est basé sur la recherche d'intersection accélérée par SIMD avec une modification, le calcul des distances ne se fait plus pour tous les plans supports mais par groupe de K plans.

3.2 Surfaces occlusives

Pour obtenir la liste des surfaces occlusives de deux surfaces non coplanaires⁴, on se base sur deux observations :

1. Tout rayon issu d'une surface A qui intersecte une surface B (A et B non coplanaire) est compris dans l'enveloppe convexe⁵ AB construite sur les sommets de ces deux surfaces (Figure 2).
2. Une surface C est potentiellement intersectée par un rayon voyageant entre A et B si cette surface C intersecte l'enveloppe convexe AB (Figure 3).

La liste des surfaces occlusives pour les surfaces A et B est donc obtenue en créant l'enveloppe convexe AB et en testant son intersection avec les $(N-2)$ autres surfaces constitutives de la salle.

³ Dénommées surfaces occlusives par la suite.

⁴ Si deux surfaces sont coplanaires, aucun rayon ne peut voyager entre elles.

⁵ L'enveloppe convexe d'un ensemble de points est le plus petit ensemble convexe qui contient ces points.

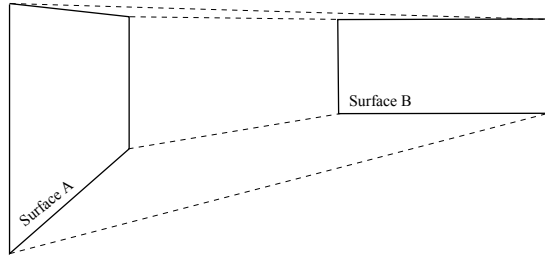


Figure 2 : Enveloppe convexe de 2 surfaces

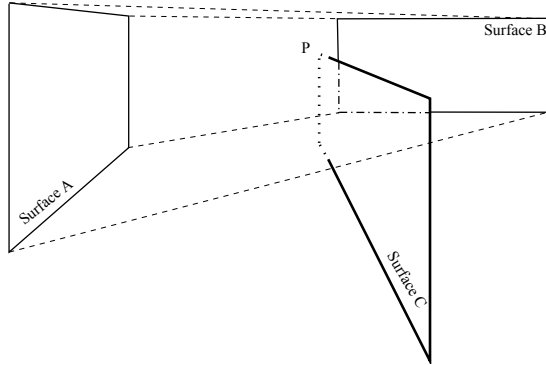


Figure 3 : La surface C est occlusive entre les surfaces A et B. P est inclus dans l'enveloppe convexe

3.3 Construction des enveloppes convexes

La construction des enveloppes convexes 3D pour toutes les paires de surfaces appartenant à la salle est réalisée à l'aide de la librairie gratuite et open source qHull [4]. On fournit l'ensemble des points de deux surfaces desquels on a extrait les doublons⁶ et on récupère la liste des facettes de l'enveloppe convexe ainsi que leur vecteurs normaux orientés vers l'intérieur du volume. Cette librairie utilise l'algorithme quickhull (3D) [6].

Puisque l'enveloppe convexe pour la paire de surfaces A-B doit être identique à celle pour la paire B-A, on ne construit qu'un nombre Q d'enveloppes convexes égal à la moitié des paires de surfaces non coplanaires. Ce qui, pour N surfaces comportant M ensembles de N_i surfaces coplanaires, donne

$$Q = \frac{N(N-1) - \sum_{i=1}^M (N_i(N_i-1))}{2} \quad (1)$$

enveloppes convexes 3D.

3.4 Recherche des surfaces ayant une intersection avec les enveloppes convexes

Une surface est potentiellement sur le chemin d'un rayon issu de la surface A à destination de la surface B si et seulement si un de ses points est inclus dans l'enveloppe convexe AB. C'est pourquoi, l'intersection de chaque surface S non coplanaire avec A ou B est testée avec l'enveloppe convexe AB.

Algorithme de test :

1. Test d'inclusion des sommets de S dans AB. Si au moins un des sommets est à l'intérieur, il y a intersection (Figure 3). Sinon, le cas est toujours indéterminé et on passe au point 2.

2. Si tous les sommets de S sont du côté extérieur d'au moins un plan support d'une face de AB, il n'y a pas d'intersection (car le volume est convexe) (Figure 4). Sinon, le cas est toujours indéterminé et on passe au point 3.
3. Si tous les sommets de AB sont du même côté du plan de support de S, alors, il n'y a pas d'intersection (Figure 5). Sinon, il y a intersection (car autrement, au moins un plan porteur d'une surface de AB aurait tous les points de S situé du côté extérieur, ce qui est impossible vu le test 2).

En cas d'intersection, cette surface est ajoutée à la liste des surfaces occlusives entre A et B.

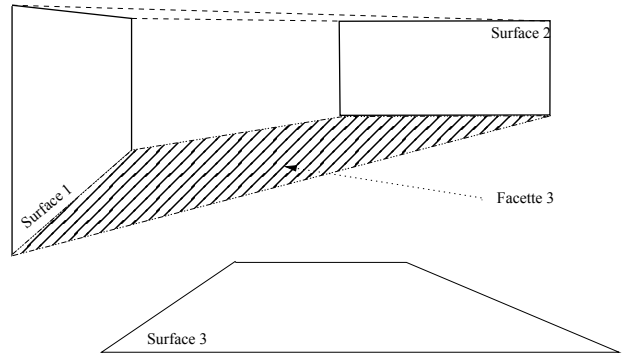


Figure 4 : Test d'intersection 2. Les sommets de la surface 3 sont tous du côté extérieur du plan de support de la facette 3

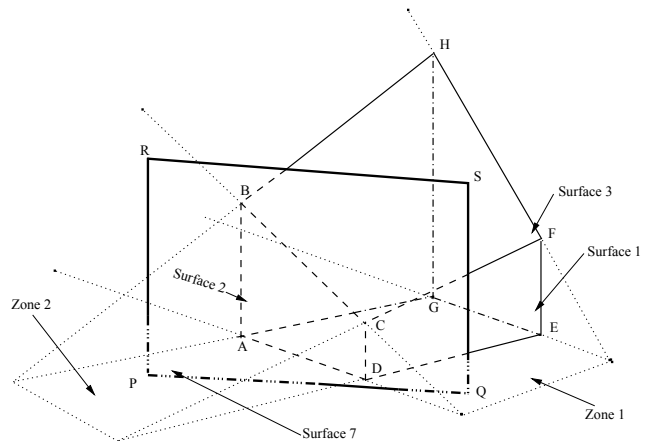


Figure 5 : Test d'intersection 3. L'absence d'intersection entre la surface 7 et l'enveloppe convexe A,B,...,H est détectée à l'étape 3 du test d'intersection

Le troisième cas de test est illustré à la « Figure 5 : Test d'intersection 3 ». L'enveloppe convexe y est délimitée par les sommets A, B, ..., H et la surface testée est la surface 7. Aucun des sommets P, Q, R, S ne sont intérieurs à l'enveloppe convexe. Le prolongement des arêtes des facettes délimite, entre autres, deux zones. Une zone 2 à l'extérieur du plan support de la surface 2 et à l'intérieur du plan de support de la surface 1 et une zone 1 dans la situation inverse. Les sommets P, Q, R, S ne sont pas tous extérieurs à au moins un plan de support des facettes de l'enveloppe convexe puisqu'il n'y a de sommets qu'à l'extérieur des surfaces 1, 2, 3 et qu'ils sont répartis des deux côtés de chacune de ces surfaces. Par contre, tous les sommets de l'enveloppe convexe sont du même côté du plan support de la surface 7.

⁶ Les doublons sont les sommets communs aux deux surfaces.

3.5 Modification des listes de surfaces occlusives

Maintenant que l'on connaît les surfaces occlusives pour toutes les paires de surfaces X Y , on pourrait les utiliser telles quelles dans le nouvel algorithme de recherche d'intersection. Cependant, des surfaces S_i peuvent être testées deux fois si, pour un segment de rayon sonore originaire d'une surface X , ces surfaces S_i :

- Ont déjà été testées négativement avant de tester la surface Y ($X \neq Y$) et,
- Sont présentes dans la liste L_{OXY} des surfaces occlusives entre X et Y . Ces surfaces seront dénommées surfaces « doublons » par la suite.

Puisqu'on connaît l'ordre de recherche dans la liste des N surfaces, on peut supprimer ces surfaces doublons des listes L_{OXY} pour chaque paire de surfaces X et Y .

3.6 Algorithme modifié

Les surfaces occlusives ne sont pas calculées entre la source et les surfaces. L'algorithme classique SIMD est appliqué aux segments de rayons issus de la source. On pourrait cependant les calculer de façon identique grâce aux enveloppes convexes et appliquer l'algorithme prenant en compte les surfaces occlusives à ces segments de rayons.

3.7 Résultats

L'accélération obtenue par cette méthode dépend de l'ordre des surfaces dans la liste globale et de la complexité de la salle⁷ (nombre de surfaces, nombre de surfaces occlusives entre chaque paire de surfaces).

Suite au surcoût des opérations nécessaires pour l'obtention et l'utilisation des surfaces occlusives, on peut arriver à un temps de calcul plus élevé que la méthode accélérée par SIMD sans surfaces occlusives.

4 Classement des surfaces

Puisque les résultats précédents dépendent de l'ordre de test des surfaces, celui ci va être organisé de façon à minimiser le nombre de surfaces testées avant d'obtenir la surface destination. Pour ce faire, il faudrait idéalement tester la surface destination en premier.

On peut se rapprocher de ce cas idéal en classant les surfaces suivant l'ordre décroissant de probabilité d'intersection avec un segment de rayon sonore. Ces probabilités dépendent de l'origine du segment de rayon et de sa direction et, bien entendu, de la géométrie de la salle. Elles sont a priori inconnues.

Une façon de les approximer, c'est d'utiliser les statistiques d'intersection des surfaces par les segments de rayons sonores issus de L rayons tirés dans les mêmes conditions que lors de la simulation finale.

Dans cette étude, on fixe deux conditions :

- La direction du rayon ne sera pas prise en compte dans le tri des surfaces.
- On ne tient pas compte avec exactitude du point d'origine du segment de rayon mais seulement de la surface dont il est originaire.

Ces conditions permettent de simplifier les listes de surfaces triées (plus le nombre de critères de classement

d'un segment augmente, plus le nombre de ces listes augmente). Pour n critères présentant chacun $\#critère_i$ valeurs possibles ($i = 1, \dots, n$), le nombre de listes est de l'ordre

$$\prod_{i=1}^n (\#critère_i) \quad (2)$$

Puisque le seul critère de tri sera la surface d'origine du segment, on aura une liste triée par surface, ce qui nous conduit à construire au total N listes L_{Ti} composées de $(N-1)$ surfaces triées et accompagnées de $(N-1)$ listes de surfaces occlusives (N est le nombre total de surfaces). Au total, on obtient $N(N-1)^2$ listes de surfaces.

4.1 Création des listes triées

Pour éviter de tirer L rayons sonores uniquement pour le tri des surfaces, on se sert des L premiers rayons de la simulation pour calculer I_{ij} , le nombre de segments de rayons originaires d'une surface S_i et ayant une intersection avec une surface S_j .

Après le tir des L rayons, on peut créer les listes de surfaces triées L_{Ti} . Dans chaque L_{Ti} , les surfaces S_j sont triées par ordre de I_{ij} décroissant.

De la même façon qu'au point 3.5, les listes de surfaces occlusives sont mises à jour en y éliminant les surfaces doublons. C'est-à-dire, si une surface X apparaît avant une surface Y dans L_{Ti} et si elle est également présente dans la liste des surfaces occlusives entre S_i et Y , on la retire de cette liste. Dès lors, lorsqu'un rayon issu de S_i possède une intersection avec Y , la surface X a déjà été traitée lors du parcours de L_{Ti} . Puisque X a été retirée de la liste des surfaces occlusives entre S_i et Y , elle ne sera pas traitée une seconde fois lors du parcours des surfaces occlusives.

4.2 Algorithme modifié

Quand les listes triées ont été créées et les listes occlusives ont été mises à jour, le tir de rayons suit l'algorithme présenté au point 3.1.

Comme au point 3.6, quand l'origine du segment de rayon est la source, l'algorithme classique SIMD est utilisé jusqu'à la première réflexion. On pourrait également utiliser une liste de surfaces triées complétées de leurs surfaces occlusives pour les segments originaires de la source.

4.3 Résultats

Le surcoût de cette méthode est compensé par la réduction des surfaces traitées lors de la recherche d'intersections si le nombre de surfaces total est suffisant, voir section suivante.

5 Simulations

Les simulations ont été réalisées sur la configuration suivante : Pentium 4, 1GB de RAM et Windows XP.

Les algorithmes suivant seront testés sur les trois salles :

1. Algorithme classique
2. Algorithme classique avec instructions SIMD
3. Algorithme des surfaces occlusives sans tri et instructions SIMD.
4. Algorithme des surfaces occlusives avec tri⁸ et instructions SIMD.

⁷ Moins il y a de surfaces testées avant de trouver la première intersection et moins sa liste de surfaces occlusives est grande, plus la recherche sera rapide.

⁸ 800 rayons sont consacrés au tri des surfaces.

5.1 Salle parallélépipédique

La salle parallélépipédique comporte six surfaces. Aucune surface n'est occlusives si les surfaces ne débordent pas. Cette salle est une longue pièce disproportionnée, utilisée par Hodgson [7].

Le fait qu'elle soit disproportionnée fournit une distribution de probabilité d'intersection non uniforme. Le faible nombre de surfaces permet de modifier aisément l'ordre initial des surfaces et d'observer l'influence de ce facteur sur le nombre de rayons traités par secondes.

Ordre des surfaces :

- RoomB1. Les grandes surfaces sol et plafond sont au milieu de la liste.
- RoomB2. Les grandes surfaces sol et plafond sont en fin de liste.
- RoomB3. Les surfaces sont classées par aire décroissante.

5.2 Salle cubique

La salle cubique possède, dans sa configuration de base, six surfaces de mêmes dimensions (configuration « Cube1 »). Si on modifie l'ordre des surfaces en échangeant les trois premières surfaces avec les trois dernières, on obtient la configuration « Cube2 ».

La salle est ensuite modifiée en ajoutant une septième surface en son centre, configuration « Cube3 ».

Si cette septième surface est divisée en deux surfaces, on obtient la configuration « Cube4 ».

5.3 Amphithéâtre

Cette salle de cours (Figure 6) est modélisée par 72 surfaces. Il existe de nombreuses surfaces occlusives et les surfaces sont débordantes.

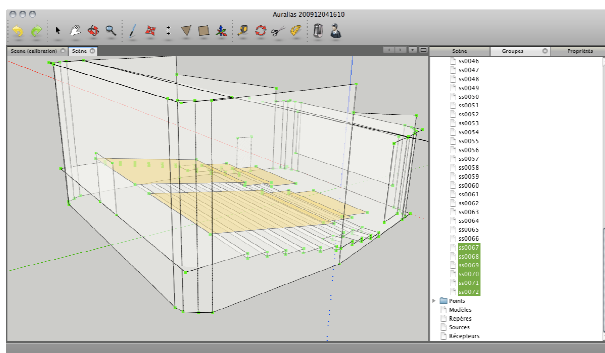


Figure 6 : Amphithéâtre composé de 72 surfaces

5.4 Résultats

Le nombre moyen de rayons tirés par seconde est donné pour chaque salle et chaque algorithme dans les Tableau 1, Tableau 2 et Tableau 3.

	Salles		
Algorithmes	RoomB1	RoomB2	RoomB3
Classique	11150	10700	11300
Classique SIMD	12800	12500	13000
Occlusions sans tri	13900	13300	14100
Occlusions avec tri	14600	14600	14600

Tableau 1 : Nombre de rayons traités par secondes pour room B

Les résultats de room B dépendent faiblement de l'ordre de parcours des surfaces (Tableau 1), y compris pour l'algorithme classique⁹. Suite à l'absence de surfaces occlusives, les méthodes 3 et 4 sont les plus performantes (Tableau 4) puisque lorsque la recherche s'arrête à la première intersection trouvée, c'est aussi la seule intersection possible. En moyenne, les algorithmes de la méthode 3 et 4 parcourent moins de surfaces.

	Salles			
Algorithmes	Cube1	Cube2	Cube3	Cube4
Classique	11060	11060	10160	9400
Classique SIMD	13250	13260	11900	12900
Occlusions sans tri	14260	14330	11660	11600
Occlusions avec tri	14660	14600	12000	11460

Tableau 2 : Nombre de rayons traités par secondes pour la salle cubique

La modification de l'ordre des surfaces de la salle cubique dans les configurations Cube1 et Cube2 est sans grande influence sur le nombre de rayons traités par seconde (Tableau 2) puisque les probabilités d'intersection des surfaces sont semblables pour la moyenne des segments de rayons.

Pour les configurations Cube3 et Cube4, l'ajout de surfaces au milieu du volume conduit à la présence de surfaces occlusives pour les algorithmes 3 et 4. Suite au surcoût introduit pour leur traitement et le très petit nombre de surfaces, ces algorithmes n'apportent pas de gain et/ou sont légèrement plus lents que l'algorithme 2 (Tableau 5).

	Salles
Algorithmes	Amphithéâtre
Classique	1400
Classique SIMD	3360
Occlusions sans tri	3000
Occlusions avec tri	4960

Tableau 3 : Nombre de rayons traités par secondes pour l'amphithéâtre

Dès que le nombre de surfaces augmente (Tableau 3), les résultats de la méthode 3 restent inférieurs à ceux de la méthode 2 (Tableau 6). Par contre, la performance de la méthode 4 devient bien supérieure à toutes les autres méthodes grâce à une meilleure organisation des surfaces.

	Salles		
Algorithmes	RoomB1	RoomB2	RoomB3
Classique SIMD	15%	17%	15%
Occlusions sans tri	24%	24%	24%
Occlusions avec tri	31%	36%	29%

Tableau 4 : Différences relatives (%) du nombre de rayons traités par seconde par rapport à l'algorithme classique pour room B

⁹ Il existe une faible dépendance à l'ordre des surfaces pour la méthode classique puisque si on trouve la surface destination avec le segment de rayon le plus court à la i^{ème} surface, on évite les tests d'inclusion « point d'intersection - surface » pour les N-i surfaces suivantes

	Salles			
Algorithmes	Cube1	Cube2	Cube3	Cube4
Classique SIMD	20%	20%	17%	37%
Occlusions sans tri	29%	29%	15%	23%
Occlusions avec tri	32%	32%	18%	22%

Tableau 5 : Différences relatives (%) du nombre de rayons traités par seconde par rapport à l'algorithme classique pour la salle cubique

	Salles
Algorithmes	Amphithéâtre
Classique SIMD	140%
Occlusions sans tri	114%
Occlusions avec tri	254%

Tableau 6 : Différences relatives (%) du nombre de rayons traités par seconde par rapport à l'algorithme classique pour l'amphithéâtre

6 Conclusion

Pour une salle comportant plusieurs dizaines de surfaces et en présence de surfaces occlusives, les simulations montrent que les performances de la recherche d'intersection basée sur le parcours partiel de la liste des surfaces sont étroitement liées à l'ordre de parcours de ces surfaces.

L'utilisation du nombre d'intersections des rayons sonores avec les surfaces pour le classement s'est avérée efficace. D'autres critères de tri pourraient être testés et évalués.

Le nombre de rayons à tirer pour classer les surfaces d'une salle de complexité moyenne telle que l'amphithéâtre est négligeable si on le compare au nombre de rayons tirés lors des simulations.

Un des points critiques de cette méthode est la génération rapide des listes de surfaces occlusives. Dans les exemples présentés, le temps passé pour cette génération reste inférieur au gain apporté par la méthode avec tri et l'implémentation pourrait encore être optimisée.

Pour conclure, pour des salles comportant quelques dizaines de surfaces, le gain de performance est important par rapport à l'algorithme classique avec ou sans utilisation d'instructions SIMD. Par contre, si le nombre de surfaces est faible et le taux de surfaces occlusives élevé, les performances peuvent être inférieures à l'algorithme classique utilisant les jeux d'instructions SIMD.

Références

- [1] Vorländer M., "Simulation of the transient and steady state sound propagation in rooms using a new combined sound particle-image source algorithm", *J. Acoust. Soc. Am.*, 86, 172-178 (1989).
- [2] Drumm I.A., Lam Y.W., "The adaptive beam tracing algorithm", *J. Acoust. Soc. Am.*, 107, 1405 (2000).
- [3] Intel, "Intel 64 and IA-32 Architectures optimization reference manual", <http://www.intel.com/Assets/PDF/manual/248966.pdf>, 2.60-2.65 (2009).
- [4] University of Minnesota, "Qhull code for Convex Hull, Delaunay Triangulation, Voronoi Diagram,

and Halfspace Intersection about a point", <http://www.qhull.org>.

- [5] Embrechts J.J., "Broad spectrum diffusion model for room acoustics ray-tracing algorithms", *J. Acoust. Soc. Am.*, 107, 2068-2081 (2000).
- [6] Bradford Barber C., Dobkin D.P., Huhdanpaa H., "The quickhull algorithm for convex hulls", *ACM Transactions On Mathematical Software*, 22, 469-483 (1996).
- [7] Hodgson M.R., "Evidence of diffuse reflection in rooms", *J. Acoust. Soc. Am.*, 89(2), 765-771 (1991).