

## La place de la radiosit  dans la qu te du r alisme virtuel

La mani re dont les ordinateurs arrivent de nos jours   simuler des images en temps r el de mani re r aliste est saisissante, et nous avons donc voulu savoir par quels proc d s des ph nom nes complexes comme la diffusion de la lumi re pouvaient  tre calcul s efficacement par ordinateur.

Le r alisme dans les jeux vid os est une probl matique au c ur de leur d veloppement. D'ailleurs, le domaine le plus complexe dans cette qu te de r alisme est, depuis longtemps, la simulation de la lumi re, en constante  volution depuis quelques ann es, et vers laquelle nous nous sommes donc orient s.

**Ce TIPE fait l'objet d'un travail de groupe.**

**Liste des membres du groupe :**

- *BA ZA Mika*

**Positionnement th matique ( TAPE 1) :**

- *INFORMATIQUE (Informatique pratique)*
- *INFORMATIQUE (Informatique Th orique)*
- *MATHEMATIQUES (G om trie)*

**Mots-cl s ( TAPE 1) :**

**Mots-cl s (en fran ais) Mots-cl s (en anglais)**

<i>radiosit�</i>	<i>radiosity</i>
<i>�clairage</i>	<i>lighting</i>
<i>rendu</i>	<i>rendering</i>
<i>simulation</i>	<i>simulation</i>
<i>moteur 3D</i>	<i>3D engine</i>

**Bibliographie comment e**

La volont  de simuler de mani re r aliste les effets de la lumi re sur un environnement gr ce   l'informatique na t   la fin des ann es 1960, mais il faut attendre les ann es 80 pour voir appara tre les premiers algorithmes de rendu d'illumination "globale", qui prennent en

comptent les effets de diffusion de la lumière entre les objets de la scène et les effets d'ombre qui en résultent. C'est en 1984 qu'apparaissent les premiers algorithmes de radiosité. Pensée initialement comme une méthode de calcul des échanges d'énergie entre des surfaces, la radiosité a été adaptée à la problématique de la modélisation de la lumière. Elle consiste à simuler itérativement les flux de lumière échangés entre chaque surface de la scène, jusqu'à atteindre un état d'équilibre. Elle se distingue ainsi des autres méthodes de rendu par sa prise en compte du principe de diffusion, qui permet d'obtenir des images réalistes où les faces colorées vont transmettre une partie de leur couleur aux faces environnantes [1].

La radiosité a été particulièrement employée dans le secteur du jeu vidéo jusqu'aux années 2000, puisque très efficace pour afficher des images pseudo-réalistes. Cependant, les calculs nécessaires à cette technique sont particulièrement coûteux, ainsi il était nécessaire de précalculer les effets de lumière des surfaces de chaque scène. De nos jours, la radiosité est rarement utilisée pour modéliser seule l'illumination d'une scène : c'est plutôt une méthode de pré-traitement utilisée sur des scènes à géométrie statique, les effets dynamiques étant gérés par des algorithmes de lancers de rayons [2].

La radiosité seule calcule la lumière au sein de la scène, mais ne permet de pas d'afficher la scène sur un écran : il faut dès lors implémenter un moteur de rendu 3D dédié, qui se chargera d'afficher la scène, représentée comme un ensemble de triangles, sur l'écran en deux dimensions, et ce sont des outils mathématiques comme les matrices de projection[3], et des algorithmes comme celui du z-buffer qui sont particulièrement utilisés dans ce domaine [4].

Au vu de la lenteur d'exécution de l'algorithme de radiosité, de nombreuses optimisations ont été imaginées pour réduire son temps de calcul [1]. En particulier, il est nécessaire de repérer de manière efficace les faces d'une scène qui ne peuvent pas interagir entre elles, ainsi que les zones d'une scène qui sont trop éloignées pour pouvoir impacter de manière significative leur luminosité respective. La structure de donnée choisie pour stocker les différentes faces de la scène est donc un choix important pour pouvoir résoudre ces problèmes efficacement, et c'est en général le choix de l'arbre k-dimensionnel qui est fait [5]. Son emploi permet en effet de déterminer rapidement l'éloignement entre plusieurs faces d'une scène, ainsi que la présence de faces obstruant potentiellement le flux lumineux de deux faces de la scène.

## **Problématique retenue**

Comment utiliser une méthode de calcul de lumière comme la radiosité de manière assez efficace pour qu'elle puisse être utilisée dans le cadre d'un jeu vidéo ?

## **Objectifs du TIPE du candidat**

- Mettre en place un algorithme naïf de calcul de la radiosité.

- Implémenter des algorithmes d'ajustement de la luminosité des pixels afin d'obtenir une image plus réaliste pour l'œil humain.
- Implémenter des structures de données permettant de détecter si une face obstrue l'échange de lumière entre deux autres faces.
- Essayer de paralléliser nos calculs pour améliorer les performances.

## Références bibliographiques (ÉTAPE 1)

- [1] MICHAEL F. COHEN ET JOHN R. WALLACE : Radiosity and Realistic Image Synthesis : [http://twanclik.free.fr/electricity/electronic/pfdone12/Radiosity%20and%20realistic%20image%20synthesis%20Cohen%20M.F.,%20Wallace%20J.R.%20\(AP,%201995\)\(412s\).pdf](http://twanclik.free.fr/electricity/electronic/pfdone12/Radiosity%20and%20realistic%20image%20synthesis%20Cohen%20M.F.,%20Wallace%20J.R.%20(AP,%201995)(412s).pdf)
- [2] MICHAEL ABRASH : Quake : a post-mortem and a glimpse into the future : <http://twimgs.com/ddj/abrashblackbook/gpbb70.pdf>
- [3] BRENDAN GALEA : The Math behind (most) 3D games - Perspective Projection : [https://www.youtube.com/watch?v=U0\\_ONQQ5ZNM](https://www.youtube.com/watch?v=U0_ONQQ5ZNM)
- [4] NICOLAS JANEY : Le Z-Buffer : <http://nicolas.janey.free.fr/ZBuffer/ZBuffer.htm>
- [5] M. HAPALA ET V. HAVRAN : Review: Kd-tree Traversal Algorithms for Ray Tracing : <https://dcgi.fel.cvut.cz/home/havran/ARTICLES/cgf2011.pdf>

## DOT

- [1] : Septembre à novembre : création du moteur 3d.
- [2] : Fin novembre : affichage des premiers rendus, sans ombres projetées.
- [3] : Décembre : migration de python à C++, accélération du calcul de C et ajout des ombres projetées et de l'ombrage adouci.
- [4] : Janvier : implémentation de l'arbre k-dimensionnel et d'un premier algorithme basé sur le principe d'une "sphère de proximité".
- [5] : Février : abandon de la sphère de proximité, implémentation de la recherche d'intersection avec l'arbre k-dimensionnel.
- [6] : Mars-avril : implémentation de la parallélisation des calculs et des techniques de post processing.
- [7] : Mai : modélisation des scènes pour la présentation du TIPE.