## **Introduction**

L’objectif des structures de données est de limiter les lectures successives et offrir un accès plus direct à l’information recherchée.

Il existe plusieurs méthodes parmi lesquelles, on peut citer : la grille, l’Octree, KD-Tree, R-Tree, Arbre BSP (Binary Space Partitioning). Une étude bibliographique permettra de mieux cernée ces différentes méthodes.

1. **structure de données (index) : Octree**

## **Définition**

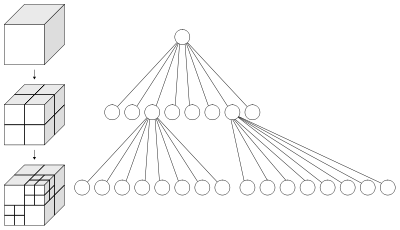
C’est une méthode de subdivision spatiale de l’espace en trois dimensions de type arbre dans laquelle chaque nœud peut compter jusqu'à huit enfants (huit octants).

## **Applications**

* Indexation spatiale
* Elimination des objets hors du cône de vue dans le cadre d'un rendu 3D
* détection efficace de collision dans le cadre de la 3D
* Ray-tracing (lancer de rayon)

## **Structure et méthode de création**

Il n’est rien d’autre qu’une structure de données. ). Le schéma ci-dessous met en évidence la relation entre le découpage de l’espace et la structure de l’arbre.



Niveau 0

Niveau 1

Niveau 2

Figure 1: (a) subdivision récursive d'un cube en octant, (b) structure de l’arbre correspondant en octree

### **Etape 1 : niveau 0 de l’arbre**

Pour créer l’Octree initial, on cherche la dimension maximum de notre scène. On ajoute enfin les dimensions pour obtenir un cube. Il s’agit donc d’un cube englobant la scène entière. On affecte par la suite, ce cube au nœud racine de l’arbre.

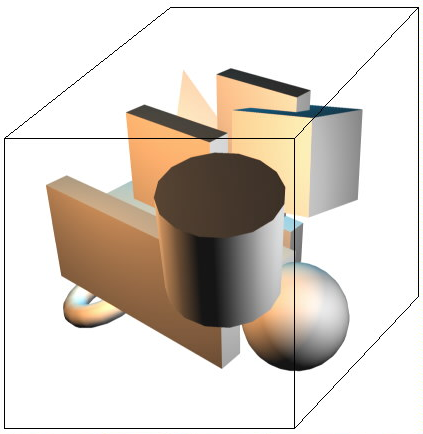


Figure 2: arbre de niveau 0 (source : http://www.flipcode.com/archives/Introduction\_To\_Octrees.shtml)

**Caractéristiques :**

* Profondeur est 0
* L’arbre contient 1 nœud
* Taille de l’octree : 1024

### **Etape 2 : niveau 1 de l’arbre**

On subdivise le cube englobant en 8 cubes plus petits à l’intérieur de celui crée à l’étape1. On obtient 4 cubes en haut et 4 cubes en bas. On ne garde que les cubes contenant au moins une surface.

**Caractéristiques**:

* Profondeur est de 1
* L’arbre contient 8 nœuds
* Nombre total octree: 8+1=9
* Taille de l’octree (bits) : 512

### **Etape 3 : niveau 2 de l’arbre**

On répète le processus de subdivision pour obtenir le niveau 2 de l’arbre. La taille des cubes est divisée par 2 par rapport au niveau précédent

**Caractéristiques**:

* Profondeur est de 2
* L’arbre contient 64 nœuds
* Nombre total octree : 9+64=73
* Taille de l’octree : 128

### **Etape 4 : niveau 3 de l’arbre**

On répète le processus de subdivision pour obtenir le niveau 3 de l’arbre. La taille des cubes est divisée par 2 par rapport au niveau précédent.

**Caractéristiques**:

* Profondeur est de 3
* L’arbre contient 512 nœuds
* Nombre total octree : 73+512=585
* Taille de l’octree : 64

### **Etape 5 : niveau 4 de l’arbre**

On répète le processus de subdivision pour obtenir le niveau 3 de l’arbre. La taille des cubes est divisée par 2 par rapport au niveau précédent.

**Caractéristiques**:

* Profondeur est de 4
* L’arbre contient 4096 nœuds
* Nombre total octree : 585+4096=4681
* Taille de l’octree : 32

On répète jusqu’à avoir une image texturée.

A l’étape 5, notre arbre possède 4681 nœuds terminaux ou feuilles

### **Cas d’arrêt**

Le processus de subdivision est récursif. On va donc se donner un temps d’arrêt. Plusieurs solutions sont envisageables :

* On arrête la subdivision lorsque le nombre de points dans le cube est plus petit qu’un seuil fixé.
* On a construit un nombre de niveau fixé à l’avance
* On arrête la subdivision lorsque la taille de notre cube à taille inférieure à une taille fixée à l’avance
* Le nombre de nœuds maximum est atteint

## **Affichage des données**

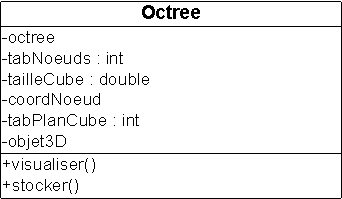
Une fois, la création de notre arbre terminée, il faut coupler à la création des points le frustrum culling. La méthode de frustrum culling permet d’afficher les points vus par le frustrum de vue. Il est assimilé au champ de vue chez l’être humain.

A voir : étudier les différentes méthodes d’optimisation

## **Analyse**

### **Diagramme cas d’utilisation (travail réalisé)**

### **Diagramme de classe**



* **Description des attributs et méthodes**

**Attributs**

octree : c’est la racine

tabNoeuds : liste d’objets pour les 8 nouveaux nœuds

tailleCube : taille du cube pour le nœud courant

coordNoeud : coordonnées X, Y, Z du nœud courant

objet3D : les objets 3D à stocker

tabPlanCube : tableau de de 6 plans du cube du nœud

**Méthodes**

**Visualiser**

**stocker**

## **Avantage et inconvénients**

### **Avantages**

* Méthode de représentation hiérarchique : visualisation multi-échelle (multi-résolution)
* Construction et exploration simples (de manière récursive)
* Méthode facile à tester les intersections avec d’autres objets
* Possibilité de visualiser les objets en 3 dimensions
* Primitive géométrique est simple : cube
* Utilisation d’un seul algorithme pour analyser et manipuler tous les objets contenus dans le cube

### **Inconvénients**

* Le rendu est coûteux en temps de calcul pour les scènes complexes
* Stockage du vide
* Nécessite beaucoup de mémoire pour le stockage d’un grand nombre de cubes
* Une représentation approximative des scènes ou des objets.

## **Bibliographie**

<http://pointclouds.org/documentation/tutorials/octree.php>

<http://recherche.ign.fr/labos/cogit/pdf/THESES/RAMOS/These_Ramos_2003.pdf>

<http://deptinfo.unice.fr/twiki/bin/view/Minfo03/AlgosJeux3DRapportOctrees>