

Étude algorithmique

MARGUERITE Alain
RINCÉ Romain

Université de Nantes
2 rue de la Houssinière, BP92208, F-44322 Nantes cedex 03, FRANCE



Table des matières

1	Stockage des boites et visualisation	2
1.1	Étude d'une solution possible : le Quadtree	2
1.2	Étude d'une seconde solution : le R-tree	3

1 Stockage des boites et visualisation

Trois problèmes majeurs apparaissent dans la réalisation du logiciel de visualisation. La première est bien entendu la gestion d'une très grande quantité de boites lors de l'affichage. Il est en effet nécessaire d'offrir un accès rapide aux informations des boites dans la fenêtre. La seconde est la gestion des filtres sur ces mêmes boites. Et le troisième apparaît lors du changement des variables étudiées (changement des dimensions visualisées).

1.1 Étude d'une solution possible : le Quadtree

Une des solutions qui permettrait d'offrir une visualisation fluide du pavage en permettant aisément de répondre au cahier des spécifications serait de représenter le pavage sous une forme de quadtree pour deux dimensions ou octree pour trois dimensions.

Le quadtree consiste à découper un espace fini en deux dimensions en quatre parties égales chacune étant stocker dans un nœud ; puis on itère ce mécanisme sur chacun de ces nœuds jusqu'à isoler spatialement les éléments recherchés. Cette structure pourrait être

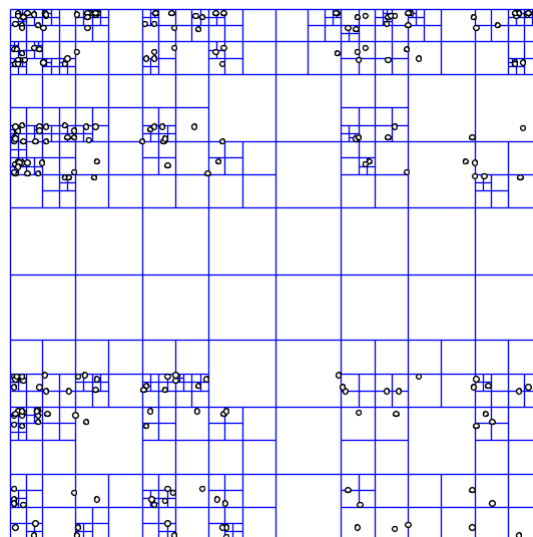


FIGURE 1 – Représentation d'un quadtree

utilisé pour déterminer la position des boites dans l'espace selon la méthode suivante :

- Si un des nœuds du quadtree ne contient aucune boite ou qu'il est entièrement inclus dans un ensemble de boites alors il n'est plus nécessaire de le subdiviser. Ce nœud contiendra donc une référence sur chacune de ces boites.
- À contrario si un nœud du quadtree contient une des bornes d'une des boites, alors il est nécessaire de subdiviser ce nœud.
- On arrête aussi de diviser les nœuds lorsque l'on arrive à une précision inférieure à la précision du calcul de Realpaver.

L'octree repose sur le même principe mais étendu à trois dimensions. L'espace est donc découpé en huit parties à chaque fois.

Cette structure est particulièrement intéressante pour la visualisation du pavage. En effet pour une fenêtre de visualisation donnée, il est très simple et rapide d'extraire la sous-arborescence correspondante à l'espace visualisé et permet aussi de ne pas afficher les objets trop petits.

Le problème de cette méthode est essentiellement à la construction puisque la structure devra découper en la plus petite feuille possible au niveau des bornes des boîtes ce qui va nécessairement entraîner la construction d'un grand nombre de feuilles. Ce problème est particulièrement gênant puisque si l'on cherche à localiser une boîte de dimensions 1 sur 1 avec la précision donnée par défaut par Realpaver de 10^{-16} , on se retrouve avec au moins 4×10^{16} feuilles. On peut optimiser l'algorithme de construction en considérant que si un des nœuds contient l'intégralité ou une partie d'une seule boîte, il n'est plus nécessaire de subdiviser, la recherche d'une seule boîte étant immédiate.

Un autre problème apparaît lorsque l'utilisateur souhaite changer les variables visualisées dans l'outil. Si l'on reste sur une structure en du type quadtree ou octree, il sera nécessaire de recalculer entièrement l'arbre.

On pourrait alors supposer une structure similaire k -dimensionnelle¹ où chaque nœud possède 2^k nœuds. L'avantage d'une telle structure est que lorsque l'on désire changer les variables de visualisation, le calcul est déjà effectué.

Malgré le fait que le quadtree pouvait être une structure intéressante, le nombre de feuilles créées est bien trop important et ne peut donc pas être utilisé.

1.2 Étude d'une seconde solution : le R-tree

Le **R-tree** est une structure de données utilisée pour stocker des informations k -dimensionnelles. Le principe est le suivant :

- Un nœud de l'arbre correspond à une boîte non-solution du pavage.
- Chaque boîte peut contenir entre m et M sous-boîtes entièrement incluses. Avec $m \leq \frac{M}{2}$.
- Une feuille de l'arbre est une boîte ne contenant que des boîtes solution du pavage.
- L'arbre est équilibré.

La figure 2 donne une bonne idée du principe des R-trees[Wik] :

Pour plus de détails sur le R-tree on pourra se référer à l'article de A. Guttman [Gut84]

Une telle structure semble bien plus intéressante en terme d'occupation mémoire. En effet le nombre de feuilles de l'arbre est au pire égale à $\frac{N}{m}$. Cependant on peut se demander si rechercher une fenêtre de visualisation sera efficace. En effet il est nécessaire

1. k étant la dimension du problème fourni à Realpaver.

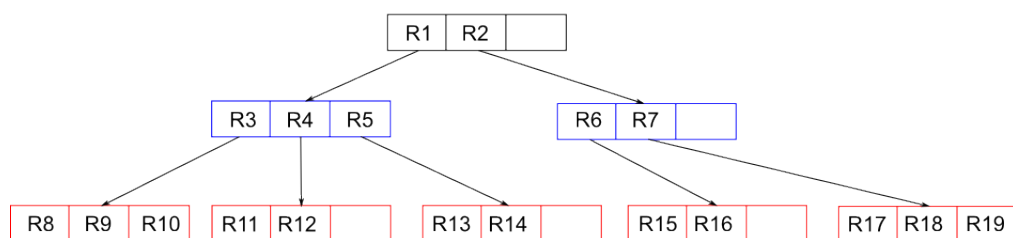
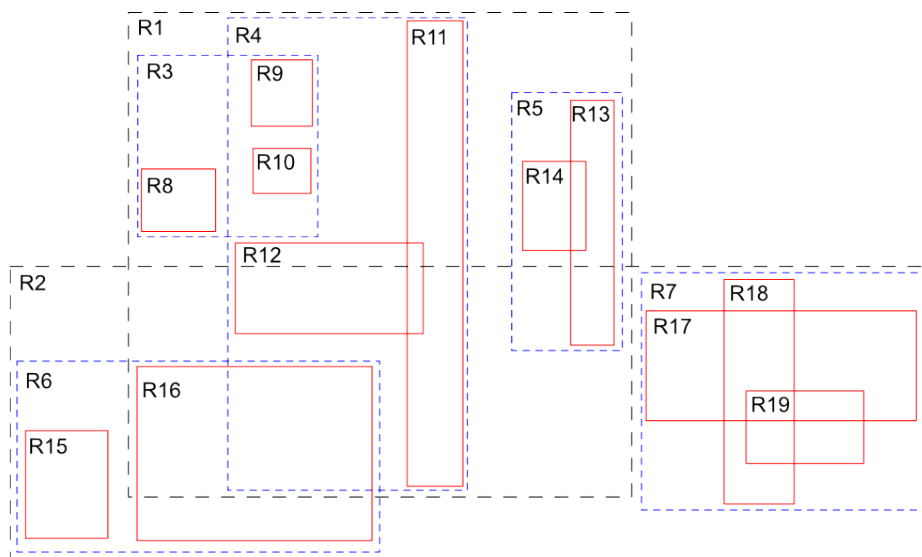


FIGURE 2 – Représentation d'un R-tree

de parcourir toutes les boites dont l'intersection est non nulle avec la fenêtre. De nombreuses analyses ont été effectuées sur le sujet. On pourra se reporter sur une analyse de performances sur les Priority R-trees [[AdBJY04](#)].

Bibliographie

- [AdBJY04] Lars Arge, Mark de Berg, Herman J.Haverkort, and Ke Yi. The priority r-tree : A practically efficient and worst-case optimal r-tree. <http://daimi.au.dk/~large/Papers/prtreesigmod04.pdf>, 2004.
- [Gut84] Antonin Guttman. R-trees : A dynamic index structure for spatial searching. <http://www-db.deis.unibo.it/courses/SI-LS/papers/Gut84.pdf>, 1984.
- [Wik] Wikipedia. R-tree. <http://en.wikipedia.org/wiki/R-tree>.