

## Partitionnement spatial grâce aux R-Arbres et application à des requêtes cartographiques.

Depuis la création en 1973 du système GPS par l'armée américaine, les systèmes cartographiques ont évolué jusqu'à devenir des outils communs. L'utilisateur formule des requêtes sur les points de la carte et compte tenu du nombre parfois gigantesque de données, il convient de trouver des stratégies accélérant de telles requêtes.

On transforme les localisations cartographiques de certains points physiques (tels que des monuments) en des coordonnées à 2 dimensions du plan, il s'agit ensuite de les organiser dans une structure informatique d'arbre afin de mener des recherches parmi ces points efficacement.

### Positionnement thématique (ÉTAPE 1) :

- *INFORMATIQUE (Informatique Théorique)*
- *MATHEMATIQUES (Géométrie)*

### Mots-clés (ÉTAPE 1) :

Mots-clés (en français)	Mots-clés (en anglais)
<i>Partitionnement de l'espace</i>	<i>Space partitioning</i>
<i>R-Arbre</i>	<i>R-Tree</i>
<i>Base de données multidimensionnelle</i>	<i>Multidimensional database</i>
<i>Plus proches voisins</i>	<i>Nearest neighbors</i>
<i>Système de navigation</i>	<i>Navigation system</i>

### Bibliographie commentée

Les requêtes de la forme « trouver les  $k$  établissements les plus proches » consistent, étant donné un ensemble de points dans un espace, à trouver le sous-ensemble des points qui sont situés à une distance (généralement euclidienne) donnée d'une origine. Ce problème se résout naïvement en un temps linéaire en le nombre de points mais l'utilisation de structures de données plus efficaces permet d'arriver à une complexité en  $O(\log(n))$ .

Le cas d'une seule dimension est bien connu et peut se résoudre grâce à un arbre binaire de recherche. La généralisation aux dimensions supérieures n'est cependant pas immédiate. L'idée d'une dichotomie en une partie "gauche" et "droite" des arbres binaires de recherche se

retrouve dans celle des Quadrees [1], structure introduite en 1974 par Finkel et Bentley, qui divise récursivement l'espace en 4 quadrants. Cette structure est déjà efficace mais la séparation systématique en quatre peut mener dans certains cas à un nombre conséquent de quadrants vides en mémoire.

Ainsi, en 1984, Antonin Guttman introduit la structure de R-Arbre [2], plus efficace en mémoire. Les feuilles de l'arbre stockent une liste de points d'intérêt, tandis que les branches contiennent une liste de nœuds, ces listes sont de longueur au plus  $M$ , constante arbitraire. Chaque nœud du R-Arbre est caractérisé par son rectangle englobant minimum : pour une feuille il s'agit du plus petit rectangle qui englobe tous les points qu'elle stocke, pour une branche il s'agit du rectangle qui englobe tous les rectangles de ses enfants. Lorsqu'on insère un nouveau point, on traverse récursivement l'arbre pour trouver la feuille la plus adaptée où l'ajouter. Dans le cas où cette feuille déborde (*i.e.* elle stocke  $M+1$  points) celle-ci est divisée en deux. Son parent peut alors aussi se retrouver avec  $M+1$  enfants et est ainsi divisé à son tour, en remontant jusqu'à la racine.

Les algorithmes proposés par A. Guttman ne sont pas les plus adaptés aux données réelles : les nœuds se chevauchent régulièrement ce qui déséquilibre l'arbre et l'algorithme de Guttman tend à créer des rectangles très fins et très allongés : bien qu'en théorie les aires de ces derniers soient optimales, ils indexent des points très éloignés, or, les requêtes cartographiques recherchent généralement des points proches entre eux. Pour résoudre ce problème, Norbert Beckmann propose en 1990 une variante, baptisée R\*-Arbre [3], munie d'un algorithme d'insertion revisité : il peut être amené à ré-insérer des objets pré-existants pour améliorer la structure de l'arbre : les rectangles des nœuds se chevauchent moins et sont moins allongés. Enfin, dernière amélioration majeure, Kamel et Faloutsos proposent en 1990 le R-Arbre de Hilbert [4] qui pré-traite les données en les triant dans l'ordre induit sur le plan par la courbe de Hilbert et améliore la construction initiale de l'arbre.

On retrouve aujourd'hui les R-Arbres dans la gestion de bases de données à plusieurs dimensions, et ils sont notamment implémentés par les principaux moteurs de bases de données telles que SQLite. Enfin, ils accélèrent les détections de collisions (par exemple en aérospatial) et permettent de faire de la synthèse d'image par tracé de rayons en temps réel. Yufei Tao propose notamment un cours sur le traitement des données à plusieurs dimensions autour des R-Arbres [5].

## Problématique retenue

Comment structurer des données spatiales éparses et sans motifs apparents en une structure d'arbre afin de mener des opérations de recherche en temps réel de façon efficace ?

## Objectifs du TIPE du candidat

1. Comprendre la structure de donnée de R-Arbre et les corrections et terminaisons des algorithmes sous-jacents.
2. Implémenter cette structure de donnée.
3. Réaliser un programme capable de répondre en temps réel à des requêtes utilisateurs cartographiques (telles que « quels sont les meilleurs restaurants dans un rayon de 5 kilomètres ? »).

## Références bibliographiques (ÉTAPE 1)

- [1] RAPHAEL A. FINKEL, JON LOUIS BENTLEY : Quad trees : A data structure for retrieval on composite keys : *Acta Informatica*, 4, (1974), 1-9
- [2] ANTONIN GUTTMAN : R-trees : a dynamic index structure for spatial searching : *Proceedings of the 1984 ACM SIGMOD international conference on Management of data*, (1984), New York, New York
- [3] NORBERT BECKMANN, HANS-PETER KRIEGEL, RALF SCHNEIDER, BERNHARD SEEGER : The R\*-tree : an efficient and robust access method for points and rectangles : *Proceedings of the 1990 ACM SIGMOD international conference on Management of data*, (1990), Atlantic City, New Jersey
- [4] IBRAHIM KAMEL, CHRISTOS FALOUTSOS : Hilbert R-tree : An improved R-tree using fractals : *VLDB*, 94, (1994), 500-509
- [5] YUFEI TAO : advanced techniques for high dimensional data - the R-tree : <https://www.cse.cuhk.edu.hk/~taoyf/course/infs4205/lec/rtree.pdf> cours à l'université de Hong-Kong, Janvier 2025