

Projet d'Algorithmique: R-trees

Andrius Ezerskis & Moïra Vanderslagmolen

April 24, 2023

Contents

1	Introduction	2
2	RTree	2
2.1	Création RTree	2
2.2	Split Linéaire	2
2.3	Split Quadratique	2
3	Structure du code	2
3.1	MBRNode	2
3.2	RTreeLinear	2
3.3	RTreeQuadratic	2
3.4	RTree	2
3.5	FileLoader	2
4	Conclusion	3
5	Bibliographie	3
6	Annexes	3

1 Introduction

2 RTree

2.1 Création RTree

2.2 Split Linéaire

Au début, nous copions l'entiereté du vecteur des enfants du node choisi pour split. Puis nous vidons ce vecteur et nous rajoutons les seeds choisies. Et puis seulement avec le vecteur copié nous ajoutons au fur et à mesure au seeds choisies. Nous nous sommes dit qu'on perdait bcp de temps à copier l'entiereté du vecteur, surtout sur des gros vecteurs.

Nous avons donc changé l'ordre, d'abord nous appelons pickNext pour qu'il rajoute les enfants aux seeds, puis nous vidons le vecteur du node à spliter et rajoutons les splits seeds.

Grâce à cette amélioration, les 8 tests que nous avons fait tourner ont pris 5,865 secondes au total au lieu de 6,778 secondes.

2.3 Split Quadratique

3 Structure du code

3.1 MBRNode

MBRNode représente les noeuds des R-Tree. Elle contient un label, un polygone, des enfants (sauf si c'est une feuille) et un parent (sauf si c'est la racine de l'arbre)

3.2 RTreeLinear

RTreeLinear représente l'implémentation du R-Tree avec l'algorithme de split linéaire.

3.3 RTreeQuadratic

3.4 RTree

RTree est une classe abstraite. Elle permet de regrouper ensemble les méthodes communes à RTreeLinear et RTreeQuadratic, par exemple la recherche d'un noeud (searchNode), l'initialisation de la classe

3.5 FileLoader

Cette classe permet de charger le fichier en mémoire. Nous avons implémenté cette classe afin de facilement changer la façon dont les fichiers sont chargés en mémoire.

4 Conclusion

5 Bibliographie

6 Annexes