

Bases de données multimédia

Séance 3 : Structuration de données pour la recherche
d'images par contenu visuel

Valérie Gouet-Brunet
valerie.gouet@cnam.fr

31 Octobre 2007



Plan de ce cours

- Problématique
- Principe général et définitions
- Une typologie des méthodes de structuration
 - Partitionnement des données
 - Partitionnement de l'espace
 - Indexation dans les espaces métriques
 - Filtrage des données
 - Linéarisation de l'espace
- Partitionnement des données : exemple du SR-tree
- Filtrage des données : exemple du VA-File
- La malédiction de la dimension

Bases de données multimédia / Structuration de données pour la recherche d'images par contenu visuel

2



Problématique

- Performances des SRI actuels
 - Matériel standard, parcours exhaustif de l'espace de description
 - Descripteurs globaux
 - Recherche encore temps-réel pour ~500 000 images
 - 500 000 descripteurs en dimension ~300
 - Descripteurs locaux
 - Recherche non temps-réel à partir de 3000 images (une image étant décrite par 300 points)
 - 900 000 descripteurs en dimension 8

Bases de données multimédia / Structuration de données pour la recherche d'images par contenu visuel 3



Problématique

- Augmentation du volume des bases d'images
 - Numérisation en masse (INA, BNF avec Gallica, etc)
 - Grand public (appareil-photos, caméscopes)
 - ...
- Descripteurs d'images de plus en plus sophistiqués
 - Descripteurs locaux
 - Descripteurs spécifiques (visages, etc)
 - ...
- Nécessité de structurer l'espace de recherche

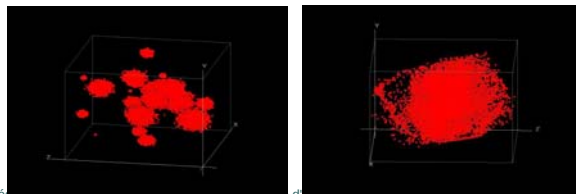
Bases de données multimédia / Structuration de données pour la recherche d'images par contenu visuel 4



Problématique

○ Problèmes

- La malédiction de la dimension
 - Au dessus de 3 dimensions, on est en grande dimension !
 - En grande dimension, des phénomènes mathématiques particulièrement non intuitifs se produisent
- La structure d'index doit être optimale pour un espace de description donné
 - Il y a bcp de distributions différentes, qui dépendent principalement :
 - Du descripteur utilisé
 - De la base d'images (notamment redondance ou non des signatures)



Bases de données multimédia / Structuration de données pour la recherche d'images par contenu visuel

5



Principe général et définitions

○ Recherche d'images par similarité visuelle

- Recherche des plus proches voisins (selon la mesure de similarité) dans l'espace de description des images

○ Taxonomie des types de requête

- Recherche par intervalle (*range query*)

$$Range(q) = \{p \in V_s / \forall i \leq d, |q_i - p_i| < s_i\}$$

- Recherche des k plus proches voisins (*k-NN query*)

$$kNN(q) = \{p \in V_s / \neg \exists r \in V_s - kNN(q), dist(q, r) < dist(q, p)\} \text{ avec } |kNN(q)| = k$$

- Recherche dans un rayon donné (*ϵ -sphere query*)

$$Sphere(q) = \{p \in V_s / dist(q, p) < \epsilon\}$$

- (Recherche approximative : compromis coût/précision)

Bases de données multimédia / Structuration de données pour la recherche d'images par contenu visuel

6



Principe général et définitions

○ Taxonomie des types de requête (suite)

● Requêtes k -NN et ε -sphere

- Comment *interpréter* et *choisir* les paramètres k et ε ?

● Interprétation de k et de ε

- k est plutôt fonction du **contenu de la base** (k = nombre de vecteurs similaires)
- ε est plutôt fonction de la **variabilité de la signature**

q et p similaires : $\text{dist}(q, p) < \varepsilon$

q et p non similaires : $\text{dist}(q, p) \geq \varepsilon$

● Choix de k et de ε

- k intuitivement facile à fixer, garantit k réponses mais risque que certaines réponses soient des images non similaires
- Problème contourné avec ε mais plus difficile à estimer (apprentissage)
- Choix dépendant du descripteur



Principe général et définitions

○ Comment accélérer la recherche ?

- Réduire les temps CPU (calcul de distances)
- Réduire les accès disque (si mémoire secondaire nécessaire)

○ Comment faire ?

● Eviter un parcours exhaustif de l'espace en le structurant

- Index multidimensionnels (structures arborescentes)
- Regroupement des données

● Filtrer l'information

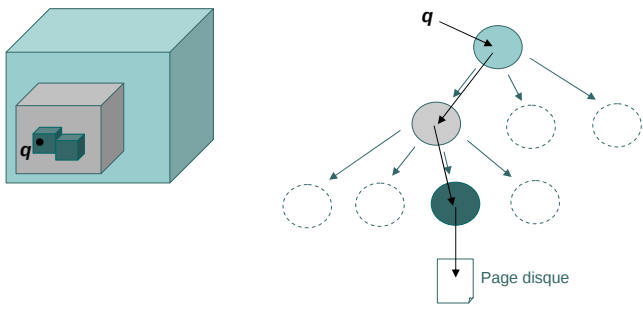
- Index multidimensionnels (quantification des données)
- Changement d'espace (projection de l'espace vers un autre plus petit)

● Faire des compromis

- Recherche approximative (compromis coût/précision)
- Réduire le nombre de dimensions de l'espace (perte d'information)

Principe général et définitions

- Les approches arborescentes traditionnelles en BD
 - Motivation
 - Réduire le nombre d'E/S et le nombre de calcul de distances
 - Principe général
 - Regroupement des données en *paquets* : hiérarchie de paquets



Bases de données multimédia / Structuration de données pour la recherche d'images par contenu visuel 9

Principe général et définitions

- Une typologie des approches de structuration
 - Partitionnement des données
 - B-tree, Famille R-tree, SS-tree, SR-tree, X-tree, ...
 - Partitionnement de l'espace de description
 - Kd-tree, LSD-tree, grid-files, ...
 - Indexation dans les espaces métriques
 - VP-tree, M-tree, ...
 - Linéarisation de l'espace de description
 - Space filling curves, Pyramid-tree, ...
 - Filtrage des données
 - VA-File, IQ-tree, LPC-File, ...

Approches *optimistes* : parcourir la plus petite partie de l'espace

Approches *pessimistes* : quitter à tout parcourir, autant le faire vite

Bases de données multimédia / Structuration de données pour la recherche d'images par contenu visuel 10



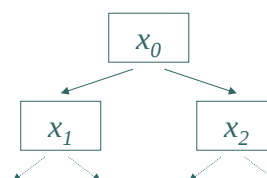
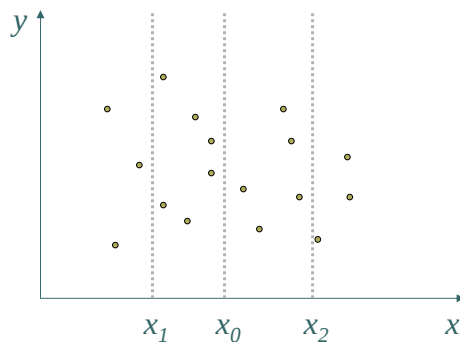
Principe général et définitions

- Une typologie des approches de structuration (suite)
 - Ces approches ont été développées pour la recherche *exacte*
 - Il existe des approches (ou des adaptations) pour la recherche *approximative* :
 - Adaptation du VA-File
 - Adaptation du M-tree
 - Regroupement de données [Berrani03]
 - SASH
 - ...



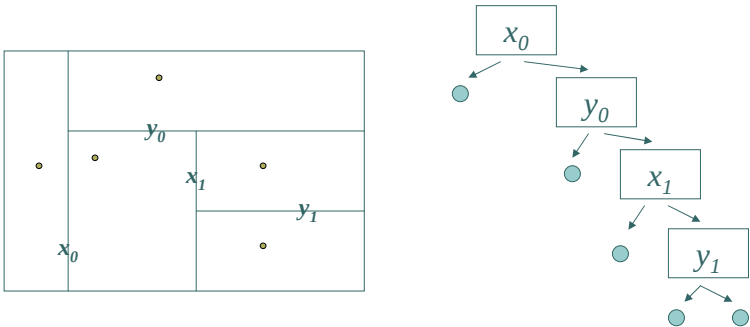
Partitionnement des données : B-tree

- B-tree : Balanced-tree (Bayer et McCreight, 1972)
 - Structurer des données selon l'une des dimensions.



Partitionnement de l'espace : Kd-tree

- Kd-tree (Bentley, 1979)
 - Arbre binaire multi-dimensionnel consistant pour chaque niveau de l'arbre à partitionner l'espace en deux sous-espaces successivement selon chaque dimension.



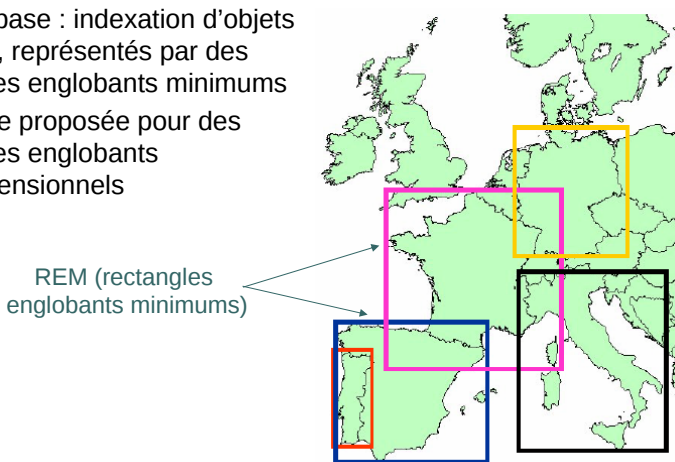
Bases de données multimédia / Structuration de données pour la recherche d'images par contenu visuel

13

Rappel : la famille R-tree

Partitionnement des données

- R-tree : Rectangle-tree (Guttman, 1984)
 - Idée de base : indexation d'objets spatiaux, représentés par des rectangles englobants minimums
 - Approche proposée pour des rectangles englobants multidimensionnels



REM (rectangles englobants minimums)

Maude Manouvrier – <http://www.lamsade.dauphine.fr/~manouvri>

Bases de données multimédia / Structuration de données pour la recherche d'images par contenu visuel

14

Rappel : la famille R-tree

Partitionnement des données

- Principe du R-tree
 - Hiérarchie de rectangles, arbre équilibré
 - Les données sont au niveau des feuilles
 - Recouvrement possible

Bases de données multimédia / Structuration de données pour la recherche d'images par contenu visuel

15

Rappel : la famille R-tree

Partitionnement des données

- Adaptation aux points multidimensionnels
 - Hiérarchie d'hyper-rectangles englobants et non disjoints, adaptée à la distribution des données

Bases de données multimédia / Structuration de données pour la recherche d'images par contenu visuel

16

Rappel : la famille R-tree

Partitionnement des données

- Un exemple de construction
 - Capacité d'un nœud : ici 3
 - Algorithme de division d'un nœud plein : minimiser la surface totale des 2 rectangles créés (split exhaustif, coût quadratique ou linéaire)

Bases de données multimédia / Structuration de données pour la recherche d'images par contenu visuel

17

Rappel : la famille R-tree

Partitionnement des données

- Recherche dans l'arbre
 - Hauteur de l'arbre
 - N : nombre de points, m : nombre minimum d'entrées par nœud
$$h \leq \lceil \log_m N \rceil - 1$$
 - Algorithme de recherche par intervalle (*range query*)
 - // Rect(X) le rectangle englobant X, Q la range query
 - Si le nœud courant est une feuille alors
 - Pour toutes les points p_i de la feuille faire
 - Si p_i est dans Rect(Q) alors
 - Placer p_i dans Liste_Résultats
 - Sinon
 - Pour toutes les fils e_i du nœud courant faire
 - Si Rect(Q) intersecte Rect(e_i) alors
 - Relancer la recherche à partir de e_i

Bases de données multimédia / Structuration de données pour la recherche d'images par contenu visuel

18

Rappel : la famille R-tree

Partitionnement des données

- Avantages/inconvénients
 - Plutôt dédié à la recherche par intervalle
 - Si dimension petite, alors un sous-ensemble réduit de nœuds est visité. Sinon, parcours séquentiel plus efficace
- Optimisations
 - Minimisation du chevauchement des rectangles englobants
 - R+-tree (1987) : n'autorise pas le recouvrement (rectangles divisés)
 - R*-tree (1990) : minimise le recouvrement et le volume des rectangles
 - X-tree (1996) : création d'un super-nœud pour mieux gérer le recouvrement
- Quelques démos (applets Java)
 - <http://donar.umiacs.umd.edu/quadtrees/rtrees/rtrees.html>
 - <http://gis.umb.no/gis/applets/rtree2/jdk1.1/help.html>

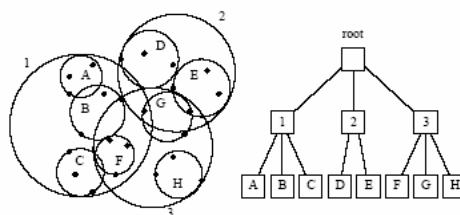
Bases de données multimédia / Structuration de données pour la recherche d'images par contenu visuel

19

Partitionnement des données : SS-tree

○ Le SS-Tree (*Similarity Search Tree*, 1996)

- Avantages/inconvénients
 - Stratégie : minimiser le diamètre
 - Plus adapté que R-tree pour la recherche par similarité



Bases de données multimédia / Structuration de données pour la recherche d'images par contenu visuel

20

Partitionnement des données : SR-tree

- L'approche SR-Tree (*Sphere/Rectangle Tree*) [Katayama et al.1997]
 - Extension du R*-Tree et du SS-Tree

Structure du R*-Tree

Structure du SS-Tree

Bases de données multimédia / Structuration de données pour la recherche

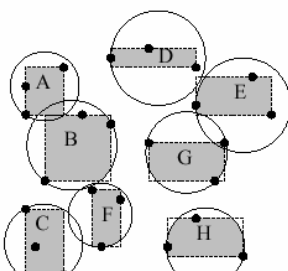
Partitionnement des données : SR-tree

- Un exemple

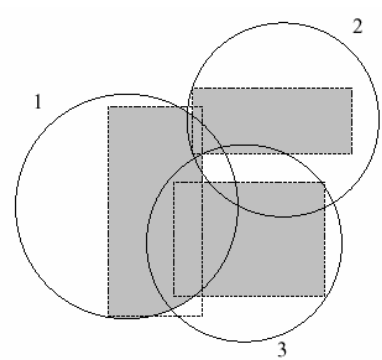
Bases de données multimédia / Structuration de données pour la recherche d'images par contenu visuel

Partitionnement des données : SR-tree

- Un exemple
 - Espace engendré par l'intersection des hypersphères et des hyperrectangles :



Niveau feuille



Niveau noeud

Bases de données multimédia / Structuration de données pour la recherche d'images par contenu visuel

23

Partitionnement des données : SR-tree

- Pourquoi combiner des hypercubes et des hypersphères ?
 - Comportement en grande dimension
 - Les hypersphères occupent un grand volume (pour un petit diamètre)

$$\text{volume}(S) = \frac{r^d \pi^{\frac{d}{2}}}{(\frac{d}{2})!} \text{ (d pair) ou } \frac{r^d \pi^{\frac{d-1}{2}} 2^{\frac{d-1}{2}}}{d!!} \text{ (d impair)}$$
 - Les hypercubes ont une grande diagonale (pour un plus petit volume)

$$\text{volume}(R) = r^d$$



- L'intersection donne un petit volume pour un petit diamètre, donc moins de recouvrement entre les nœuds

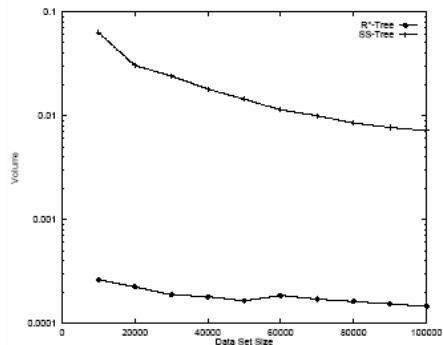
Bases de données multimédia / Structuration de données pour la recherche d'images par contenu visuel

24

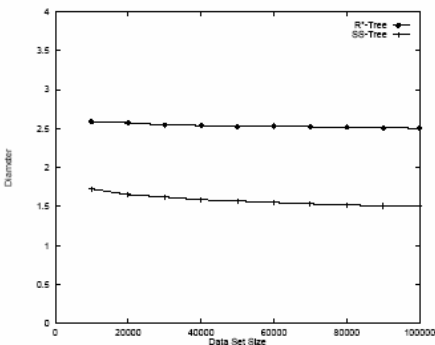


Partitionnement des données : SR-tree

- Comparaison des volumes et diamètres au niveau des feuilles



(a) Volume



(b) Diameter



Partitionnement des données : SR-tree

- Pourquoi combiner des hypercubes et des hypersphères ?

	R*-Tree	SS-Tree	SR-Tree
Region Shape			
Tree Construction Strategy	Minimizing Volume and Overlap	Minimizing Diameter	Minimizing Diameter
Diameter*	4	1	1
Volume*	60	10 ⁹	1

Espace de dimension 16



Partitionnement des données : SR-tree

○ Algorithme de recherche k-NN

// Résultats = liste triée des k plus proches voisins

Choisir k points et les classer dans Résultats

Si le nœud est une feuille alors

Pour toutes les points p_i de la feuille faire

Si $\text{distance}(\text{requête}, p_i) < \text{distanceMax}(\text{Résultats})$ alors

Placer p_i dans Résultats

Sinon

Pour toutes les entrées e_i du nœud faire

Si $\text{DistanceMin}(\text{requête}, e_i) < \text{distanceMax}(\text{Résultats})$ alors

Relancer la recherche à partir de e_i



Partitionnement des données : SR-tree

○ Adaptation à la recherche *sphere query*

Algorithm 1: $S(T, q)$

Input : T a tree root, q a sphere query (sphere with center p and radius ϵ)

Output: V a set of vectors $\langle V_1, V_2, \dots, V_m \rangle$

// Computes the set V of vectors of tree with rootpage T inside the sphere q

$V := \text{empty};$

readpage T ;

if T is not a leaf then

for each child C in T do

if $C.R$ intersects q and $C.S$ intersects q then

// $C.R$ ($C.S$) minimum bounding rectangle (sphere) of C

$V += \{S(C, q)\};$

end for;

else

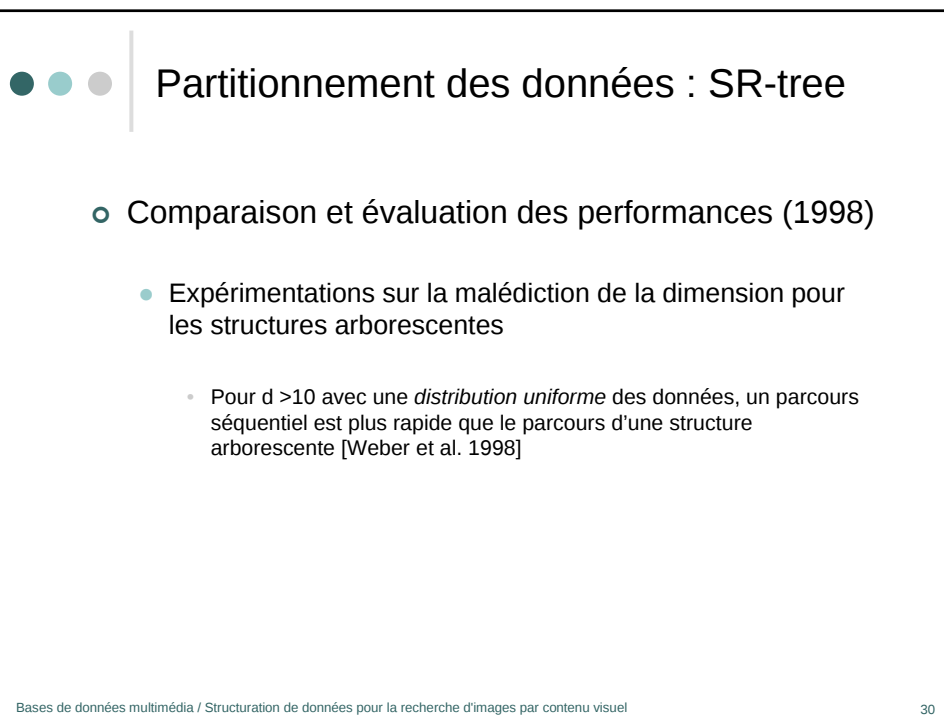
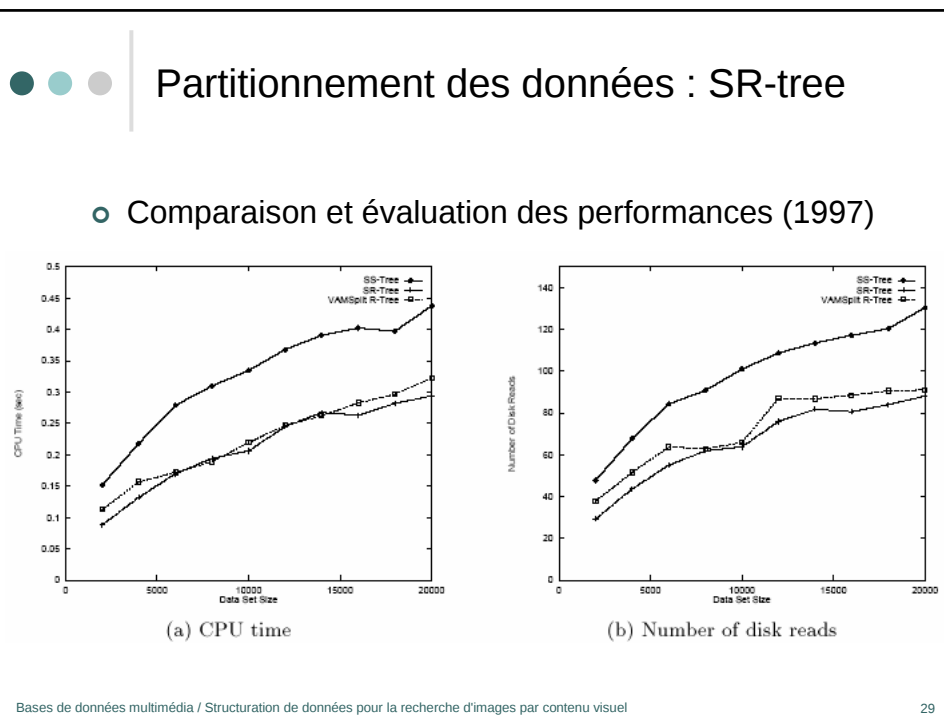
for each point t in T do

if t in q then $V += \{t\};$

end for;

end if;

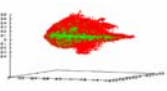
return V ;



Partitionnement des données : SR-tree

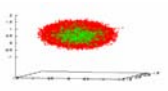
- Comparaison et évaluation des performances (2006)
 - Evaluation pour les distributions de points d'intérêt [Bouteldja et al. 2006]
 - 15 bases testées
 - 1 jeu réel (COIL-100) 
 - 4 bases synthétiques: 1 uniforme, 3 en clusters (3 variances)
 - 3 dimensions (8, 17 et 29 pour chaque base)
 - 218.000 points (conclusions confirmées avec d'autres bases de 1M de points)

COIL-100 dataset



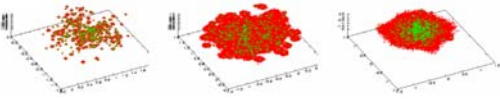
D_U^1

Synthetic uniform dataset



D_U^1

3 synthetic clustered datasets

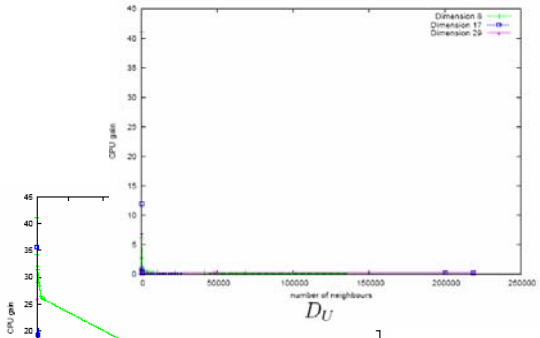


bases de données multimédia / Structuration de données pour la recherche d'images par contenu visuel

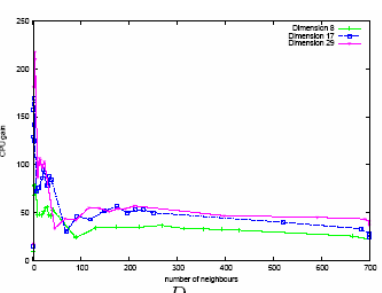
31

Partitionnement des données : SR-tree

- Comparaison et évaluation des performances (suite)
 - Gain par rapport à un parcours séquentiel



D_U



D_{C2}

à recherche d'images par c

Le SR-tree

Partitionnement des données

- Comparaison et évaluation des performances (suite)
 - Quelques caractéristiques

Arbre	Base	Vol disque	Nds internes	Nds feuilles	Supernœuds	Maximum bloc
XTREE	COIL8	17272832	201	3978	9	5
XTREE	COIL17	37482496	1017	8132		
XTREE	COIL29	78684160	4324	14884		
SRTREE	COIL8	15511552	256	3530	hauteur 4	
SRTREE	COIL17	33394688	1181	6971	hauteur 6	
SRTREE	COIL29	67129344	4392	11996	hauteur 9	

153792 points

Arbre	Base	Vol disque	Nds internes	Nds feuilles	Supernœuds	Maximum bloc
XTREE	VIDEO9	130433024	1277	29405	275	35
XTREE	VIDEO18	272080896	7985	58395	11	5
XTREE	VIDEO30	547532800	29644	104029		
SRTREE	VIDEO9	114257920	2128	25766	hauteur 5	
SRTREE	VIDEO18	234860544	8240	49098	hauteur 7	
SRTREE	VIDEO30	444874752	28775	79836	hauteur 10	

1M points

Bases de données multimédia / Structuration de données pour la recherche d'images par contenu visuel 33

Le SR-tree

Partitionnement des données

- Comparaison et évaluation des performances (suite)
 - Comparaison avec le X-tree : temps CPU

8

17

29

9

18

30

Bases de données multimédia / Structuration de données pour la recherche d'images par contenu visuel 34

Le SR-tree

Partitionnement des données

- Comparaison et évaluation des performances (suite)
 - Comparaison avec le X-tree : temps IO

8

17

29

Bases de données multimédia / Structuration de données pour la recherche d'images par contenu visuel

35

Le SR-tree

Partitionnement des données

40.745 images, 20 dimensions (histogramme couleur réduit)

<http://research.nii.ac.jp/~katayama/homepage/research/srtree/demo/>

Bases de données multimédia / Structuration de données pour la recherche d'images par contenu visuel

36

Filtrage des données : VA-File

- L'approche VA-File (*Vector Approximation File*) [Weber et al. 1997]
 - Filtrage des données par *approximation en mémoire centrale*
 - Recherche séquentielle et exhaustive *dans l'espace de recherche approché*
 - Accès disque à un *sous-ensemble* de données

Bases de données multimédia / Structuration de données pour la recherche d'images par contenu visuel 37

Filtrage des données : VA-File

- Approximation des données
 - Quantification des données en hyper-rectangles
 - Un point est approché par la cellule qui le contient

2^{b_j} tranches par dimension j
 2^b hyper-rectangles

$$b = \sum_{j=1}^d b_j$$

ID	Coordonnées
1	0.52 0.94
2	0.70 0.96
3	0.13 0.13
4	0.11 0.20
5	0.24 0.12

1	1011
2	1110
3	0110
4	0001
5	0100

VA-File

Bases de données multimédia / Structuration de données p

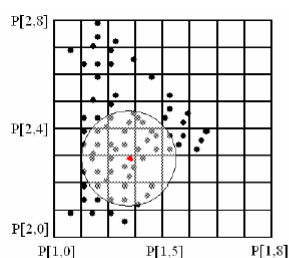


Filtrage des données : VA-File

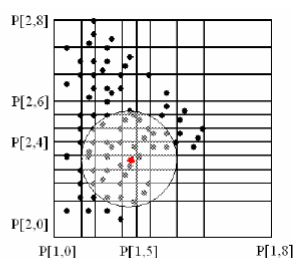
○ Approximation des données

● Construction des hyper-rectangles

- Distribution uniforme (non adapté aux données non uniformes)
- Distribution adaptative tenant compte de la distribution des points sur chaque dimension (VA*File)



(a) Partage uniforme de l'espace



(b) Partage adaptatif de l'espace



Filtrage des données : VA-File

○ Statistiques sur le nombre de cellules occupées

- Données uniformes, volume $[0..1]^d$
- $P[\text{point dans une cellule}] = \text{volume}(\text{cellule}) = 2^{-b}$
- $P[\text{une cellule contienne plus d'un point}]$

$$= 1 - (1 - 2^{-b})^{N-1} \approx \frac{N}{2^b}$$



Filtrage des données : VA-File

○ Algorithme de recherche *sphere query*

Pour chaque cellule faire

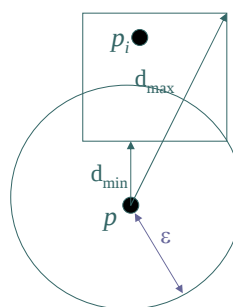
Si $\varepsilon > d_{max}$ alors

les points de la cellule sont retenus

Sinon si $\varepsilon < d_{min}$ et si $p \notin \text{cellule}$ alors

les points de la cellule ne sont pas retenus

Sinon il faut calculer $\text{distance}(p, p_i)$



Filtrage des données : VA-File

○ Algorithme de recherche k-NN

Insérer k points p_i dans une liste triée selon leur distances
 $\text{distance}(p, p_i)$

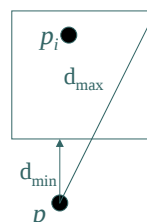
// D = plus grande distance de cette liste

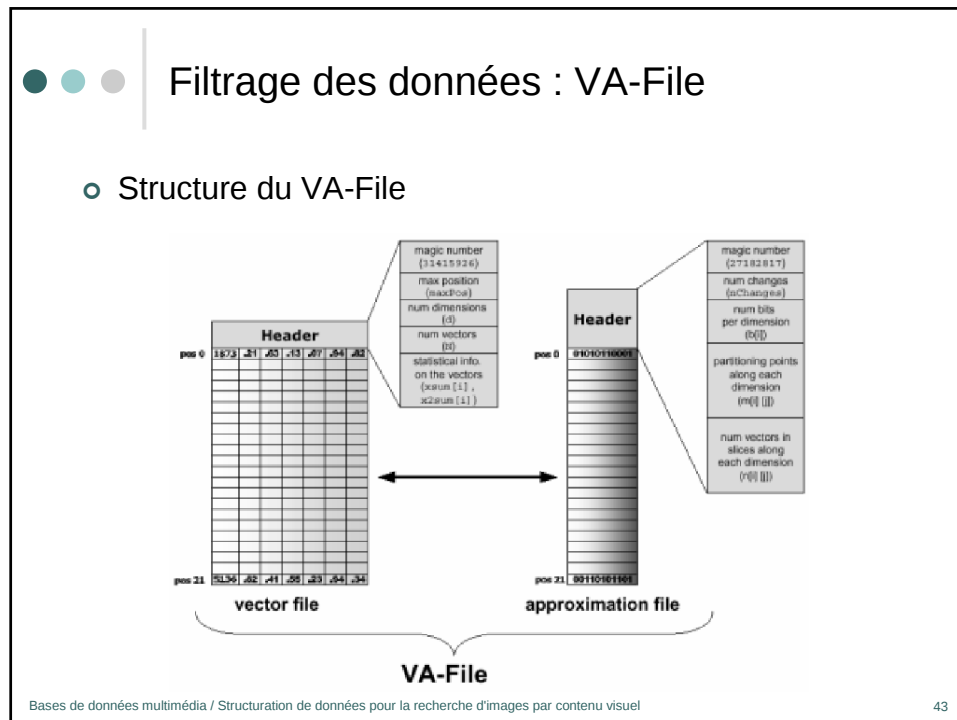
Pour toutes les autres cellules faire

Si $d_{min} < D$ alors

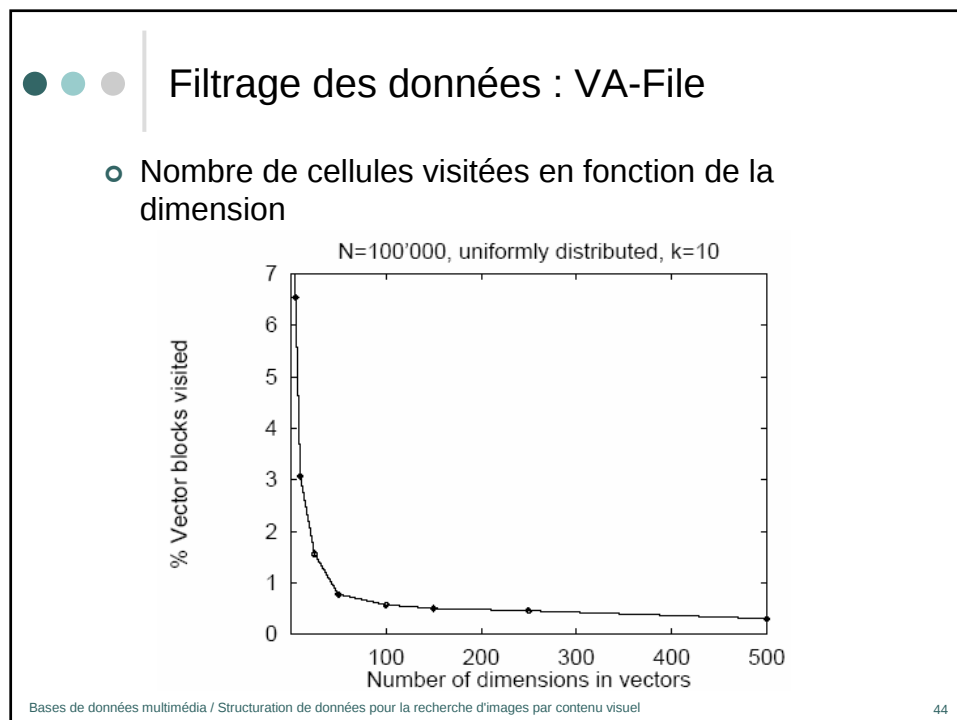
Remplacer les points de la liste par tout
ou partie de ceux de la cellule

$D \leftarrow \min(d_{max}, D)$





43



44



Filtrage des données : VA-File

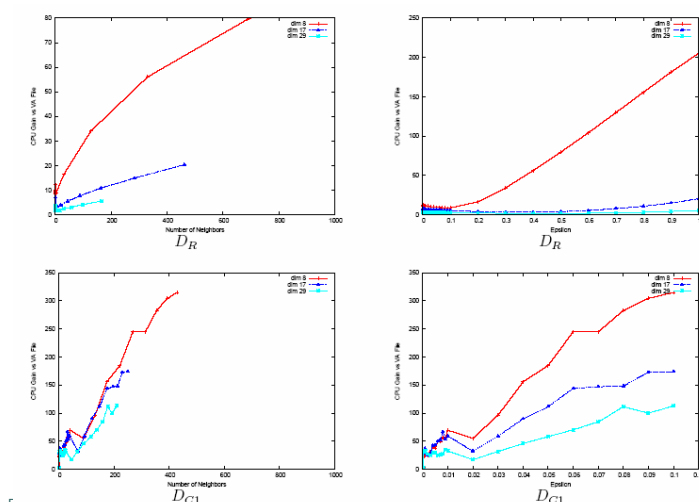
Avantages/inconvénients

- A partir de 6 dimensions pour une distribution uniforme, gain par rapports aux structures arborescentes (R*tree, X-tree)
- Si l'approximation ne tient pas en mémoire centrale, les performances deviennent inférieures à un parcours séquentiel
- L'approche n'a pas d'intérêt si les données tiennent en mémoire centrale

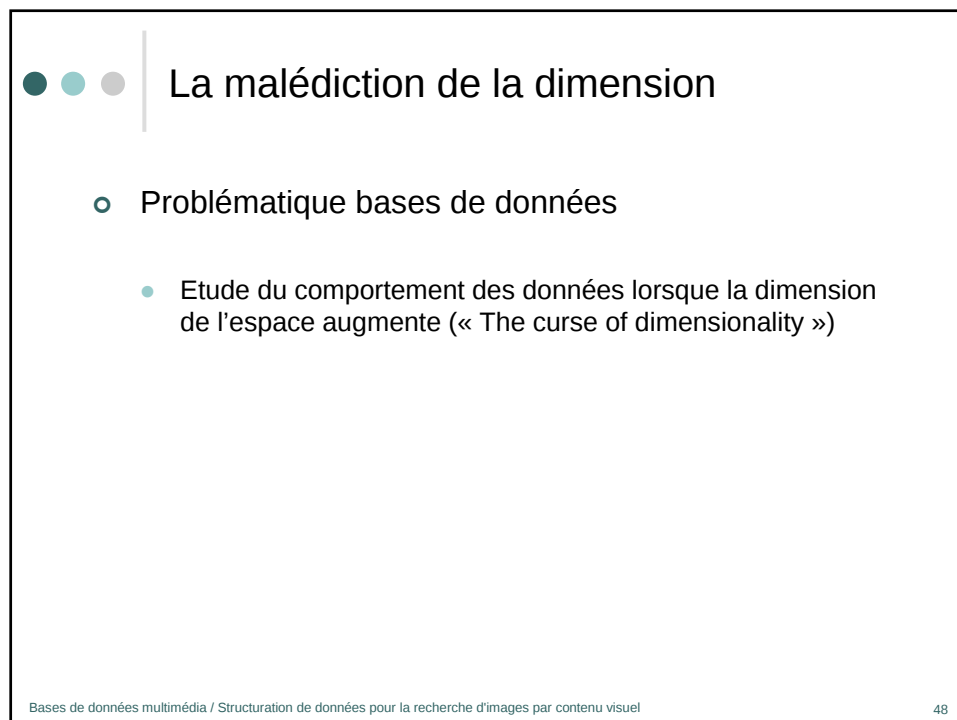
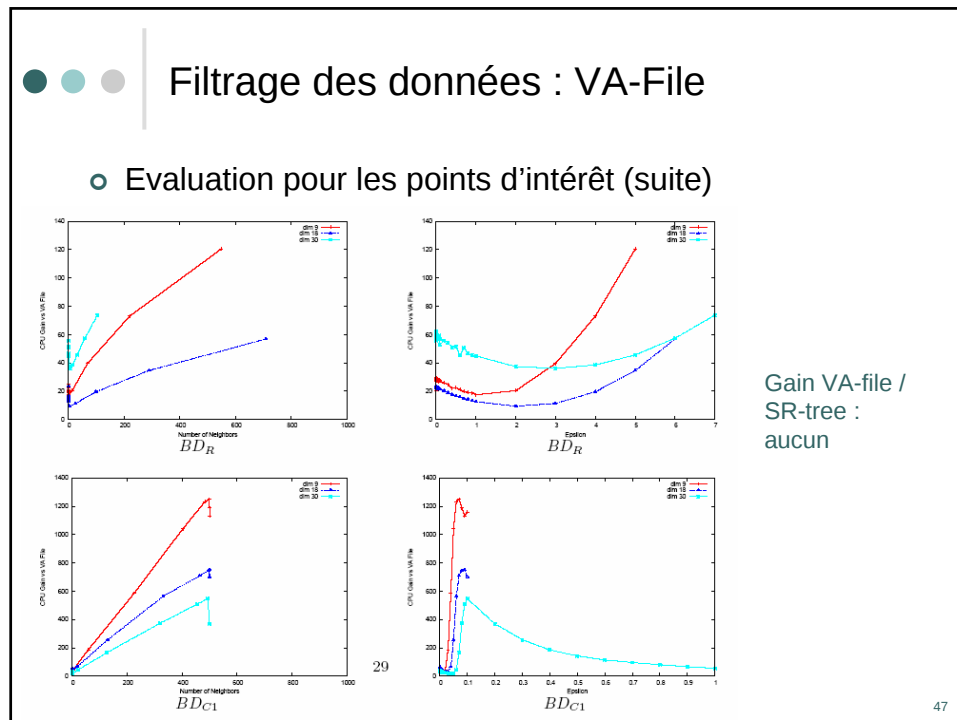


Filtrage des données : VA-File

Evaluation pour les points d'intérêt (2006)



Gain VA-file /
SR-tree :
aucun



La malédiction de la dimension

Quatre observations (hypothèse distribution uniforme)

1. Phénomène de l'espace vide

L'espace est faiblement occupé

Exemple d'une partition simple : découpage régulier en 2^d cellules

$N = 10^6, 10^{30}$ cellules pour $d=100$.

➤ Presque toutes les cellules sont vides

2. Croissance exponentielle de la taille de la base

S l'hypersphère de rayon 0,5 (plus grande hypersphère de l'espace) et N taille de la base pour qu'au moins un point en moyenne dans S :

$d = 10, N = 400$

$d = 20, N = 40 \cdot 10^6$

$d = 40, N = 3000 \cdot 10^{20}$

$d = 100, N = 5000 \cdot 10^{69}$

Bases de données multimédia / Structuration de données pour la recherche d'images par contenu visuel

49

La malédiction de la dimension

Quatre observations (suite)

3. Phénomène de concentration des mesures

En grande dimension, les distances deviennent statistiquement identiques : dans ces conditions, le bruit provenant de l'acquisition des données perturbe trop les métriques habituelles

4. Les points sont proches des (d-1) surfaces

$P[\text{point dans un cube de côté } a] = a^d$

Si $d = 100, a = 95\%, P = 0,59\%$

0.95

Dimension	Rapport des Volumes
1	1
2	0.78732
4	0.329707
6	0.141367
8	0.0196735
10	0.00399038

Rapport des volumes hypersphères / hypercubes

Bases de données multimédia / Structuration de données pour la recherche d'images par contenu visuel

50

La malédiction de la dimension

- Finalement, qu'est ce qu'un espace de grande dimension ?
 - Pour certains, c'est un espace non visualisable
 - $d > 3$
 - Dans le cadre de l'indexation multidimensionnelle
 - C'est un espace dans lequel un parcours séquentiel est plus performant que le parcours d'une structure arborescente
 - $d > 10$ (démonstration expérimentale [Weber et al. 1998])
 - Phénomène de concentration des mesures
 - $d > 20$ (démonstration mathématique [Beyer et al. 1999])
 - Proposition de [Verleysen, 2003]
 - Mettre en rapport la *dimension* avec la *taille* de la population de l'espace
 - Un espace est de grande dimension (ou de petite taille) quand la taille de sa population n'augmente pas exponentiellement avec sa dimension
 - Descripteurs globaux, espace de grande dimension ?
 - Descripteurs locaux, espace de petite dimension ?

Bases de données multimédia / Structuration de données pour la recherche d'images par contenu visuel

51

Bibliographie

- Norio Katayama and Shin'ichi Satoh, « The SR-tree: An Index Structure for High-Dimensional Nearest Neighbor Queries », *Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data*, May 1997, pp. 369-380.
- Roger Weber, Stephen Blott, « An Approximation-Based Data Structure for Similarity Search », *Technical Report 24, ESPRIT project HERMES (no. 9141)*, October 1997.
- Roger Weber, H.J. Schek and S. Blott, « A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces », *Proceedings of the 24th VLDB Conference*, New York, USA, 1998.
- Christian Böhm, Stephan Berchtold and D.A. Keim, « Searching in High-dimensional Spaces – Index Structures for Improving the Performance of Multimedia Databases », *ACM Computing Survey*, 33(3): 322-373, 2001.
- Valérie Gouet-Brunet, « Recherche par contenu visuel dans les grandes collections d'images », *Encyclopédie de l'Informatique Vuibert*, 2006
- N. Bouteldja, V. Gouet-Brunet and M. Scholl, Evaluation of strategies for multiple sphere queries with local image descriptors, *IS&T/SPIE Conference on Multimedia Content Analysis, Management and Retrieval*, San Jose CA, USA, 2006.
- M. Verleysen, Learning high-dimensional data, in *Limitations and future trends in neural computation*, pages 141-162, IOS Press, 2003.
- Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, Uri Shaft, When Is Nearest Neighbor Meaningful?, *International Conference On Database Theory*, pages 217-235,

Bases de données multimédia / Structuration de données pour la recherche d'images par contenu visuel

52