

情報学科CSコース情報システム(3年後期)

講義ノート —第9回—

空間アクセス法 (Z-ordering, R木)

本講義ノートは、以下の文献を引用
Philippe Rigaux, Michel Scholl, Agnes Voisard,
“Spatial Databases with Application to GIS”,
Morgan Kaufmann Publishers, 2002

空間アクセス法の応用

伝統的な関係DB

- k個の属性からなるレコードはk次元空間の点

地理情報システム (GIS: Geographic Information System)

医療用画像DB

- 3次元MRI脳スキャン画像

マルチメディアDB

- 多次元オブジェクトは特徴ベクトル空間での点

時系列データ解析

- k個の連続した点列はk次元空間の点

履歴DBや時制DB (Historical/Temporal DB)

- 1つのイベントは時間区間や2次元長方形

空間アクセス法

(Spatial Access Methods: SAMs)

空間アクセス法

- 多次元の点, 線, 多角形などの幾何学的なものを扱う
- 空間充填曲線 (space-filling curve, z-ordering, 線形quadtrees)
- R木とその変形

空間オブジェクトの検索

点や領域を指定し該当する空間オブジェクトを求める. 範囲質問, 近傍質問, 空間結合

範囲質問

- 指定した領域とintersect (contain) する空間オブジェクトを求める.
「京都大学から100km以内にあるすべての市や河川」

近傍 (nearest neighbor) 質問

- 指定した点・領域に最も近い空間オブジェクトを求める.
「京都大学に最も近い郵便局を3つ求めよ」

空間結合 (spatial join)・オーバーレイ

- 指定した領域と空間オブジェクトで関連あるものを結合
「この湖から10km以内のすべての市を求めよ」

空間充填曲線(z-ordering)

正方形の空間(イメージ)

イメージは $2^K \times 2^K$ 個の小正方形(ピクセル)に分割。
Kはビット数。

イメージ内の小正方形(ピクセル)に全順序

- X座標 00, 01, 10, 11, Y座標 00, 01, 10, 11
- Shuffle関数: X座標の2桁目, Y座標の2桁目, X座標の1桁目, Y座標の1桁目
- ピクセル(00,11) → 0101 → 5 (10進法)
ピクセル(00,01) → 0001 → 1 (10進法)

空間充填曲線(Z-ordering)

2次元空間でK=2の場合

領域C Z値

(01,00) → 0010 → 2

(10,00) → 1000 → 8

領域B

(10,10) → 1100 → 12

(10,11) → 1101 → 13

(11,10) → 1110 → 14

(11,11) → 1111 → 15

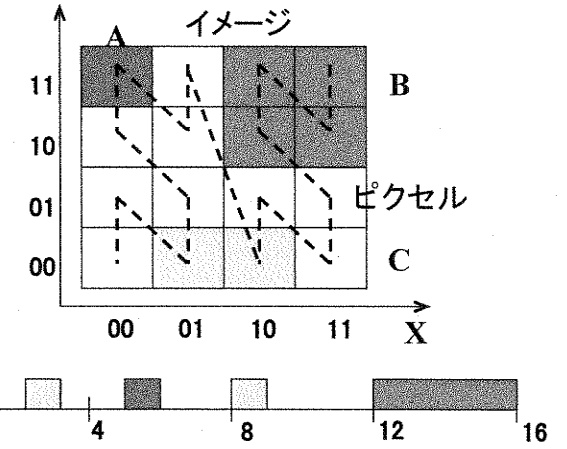
索引

(Z値, 領域名)をB+木

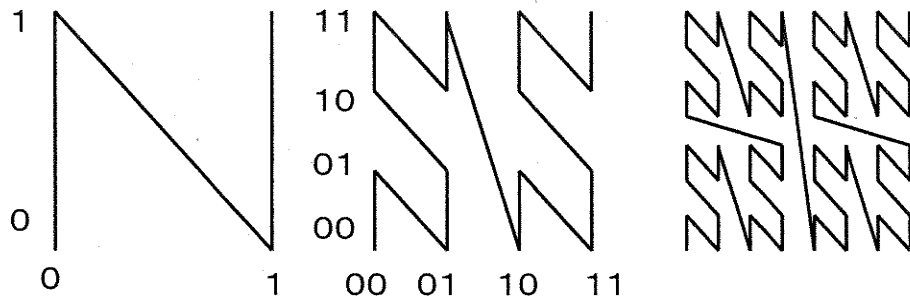
で表現. Z値がキー

質問

(10,11) → 1101 → 13



空間充填曲線(Z-ordering)



Z-order curves for 2 x 2, 4 x 4, and 8 x 8 grids

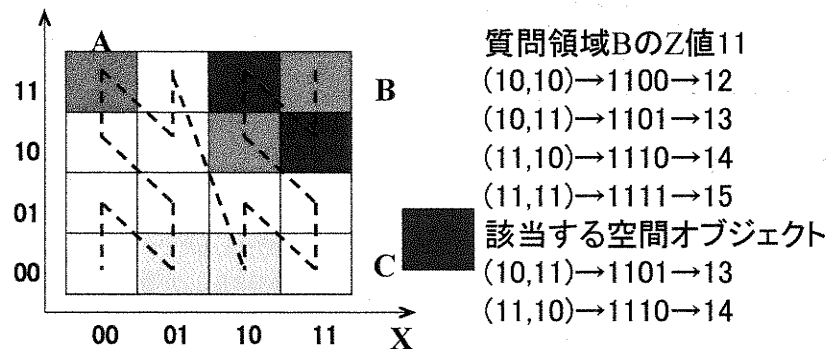
Why Z-ordering?



Z-orderingによる質問処理(1)

範囲質問

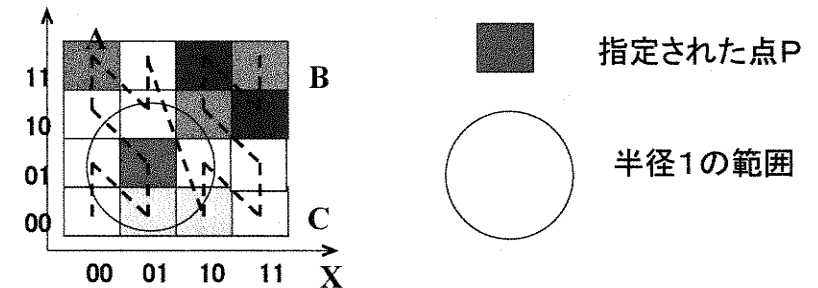
- 指定された領域→対応するピクセルのZ値の集合



Z-orderingによる質問処理(2)

近傍質問

- 指定された点P→対応するピクセルのZ値→最も近いZ値をもつ空間オブジェクトを探索→質問点との距離 r を計算→点Pを中心とする半径 r の範囲質問を実行



Z-orderingによる質問処理(3)

空間結合

- 「湖を横切るすべての鉄道を求める」
- 湖→対応するZ値の集合→ソート
 鉄道→対応するZ値の集合→ソート
 2つのソートされたZ値集合をマージ

R木(R-Tree)

Antonin Guttman (UCB), 1984年提唱.

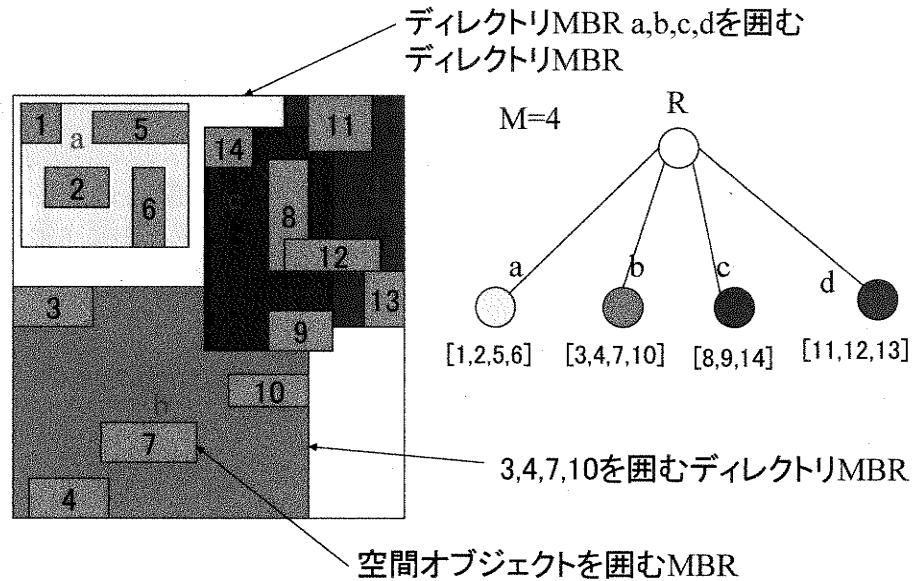
概要

- B木を n 次元オブジェクトに拡張. 平衡木.
- 空間オブジェクト(2次元領域など)は, それを含む最小の長方形 (MBR, Minimum Bounding Rectangle) で表現.

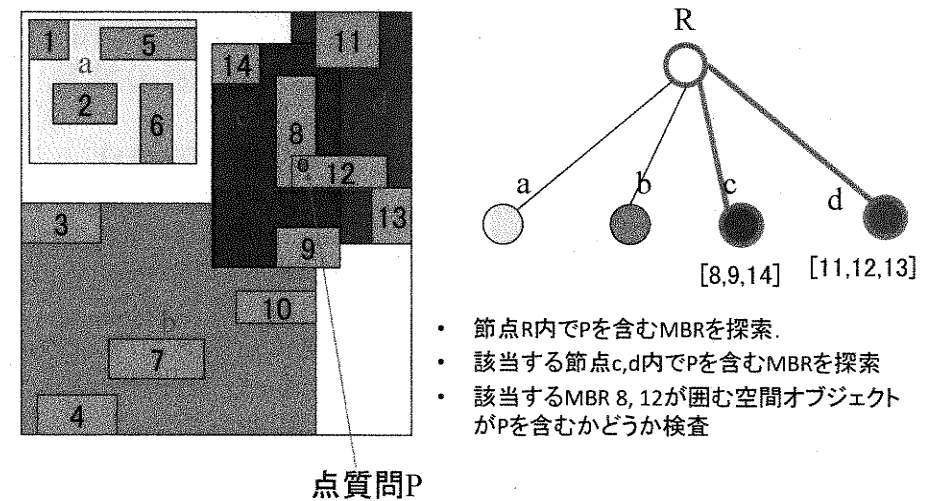
$$(x_1, y_1, x_2, y_2, \dots, x_n, y_n)$$

- R木の節点→ディスクページ
- 葉節点: m 個以上, 最大 M 個のレコード
 (オブジェクトのMBR, オブジェクトid) $m \in [0, \frac{M}{2}]$
- 非葉節点: m 個以上, 最大 M 個のレコード
 (子節点内のすべてのMBRを含むMBR, 子節点へのポインタ)
- 根節点: 最小2個のレコード

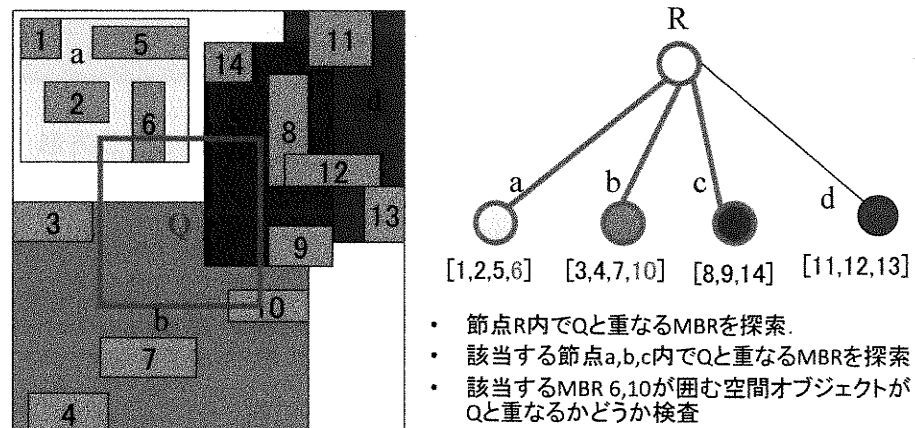
R木の例



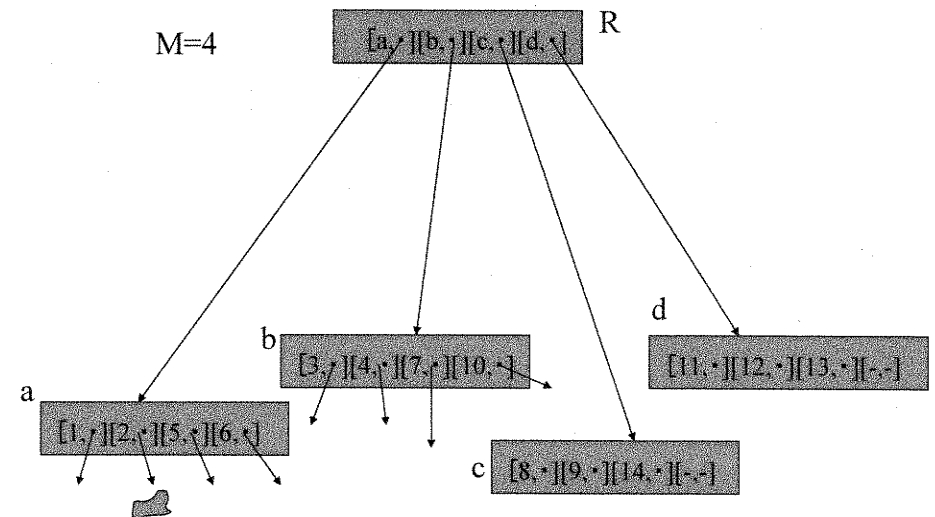
点質問(point query)とR木の例



範囲質問(range query)とR木の例



R木の具体例



R木への追加

ある空間オブジェクト o を追加

- オブジェクト o を囲むMBRを $mbr(o)$ とする。オブジェクト o のidを $oid(o)$ とする。レコード $(mbr(o), oid(o))$ をR木に追加。
- 根節点から順次探索。非葉節点中のレコード x で、 $mbr(mbr(o), x.mbr) - x.mbr$ が最小となる x を順次選択。
- 葉節点 v に到達すると、 v に空きがあれば、レコード $(mbr(o), oid(o))$ を追加。 v を指すレコードの mbr を修正する (順次上方向に向かってこれを繰り返す)
- 葉節点 v が満杯の場合、 v を2分割し、新たに v, v' とし、元の v の内容を分配する。新たな v を指すレコードの mbr を修正。また、 v' を指すレコードを v の親節点に挿入。

面積総和最小の分割

Brute-force method (風漬し法)

- あらゆる組み合わせを生成し面積総和最小のものを求める。

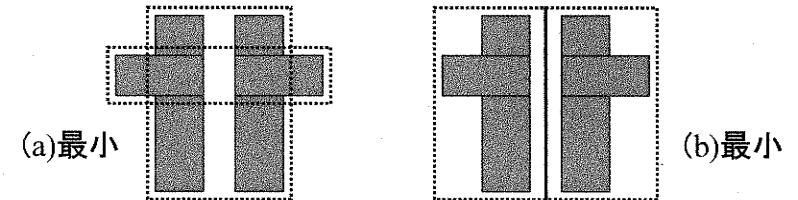
節点の分割(split)

節点 v ($M+1$ 個のレコード) の v, v' への分割

- 一方: $m+i$ 個のレコード $0 \leq i \leq M-2m+1$
- 他方: $M+1-m-i$ 個のレコード

目標: 下記の両方は不可能なので(a)を採用)

- (a) 各節点に対応するmbrの面積の総和最小
- (b) 各節点に対応するmbrの重なる部分の面積最小



面積総和最小の分割

Quadratic split (2乗分割) アルゴリズム

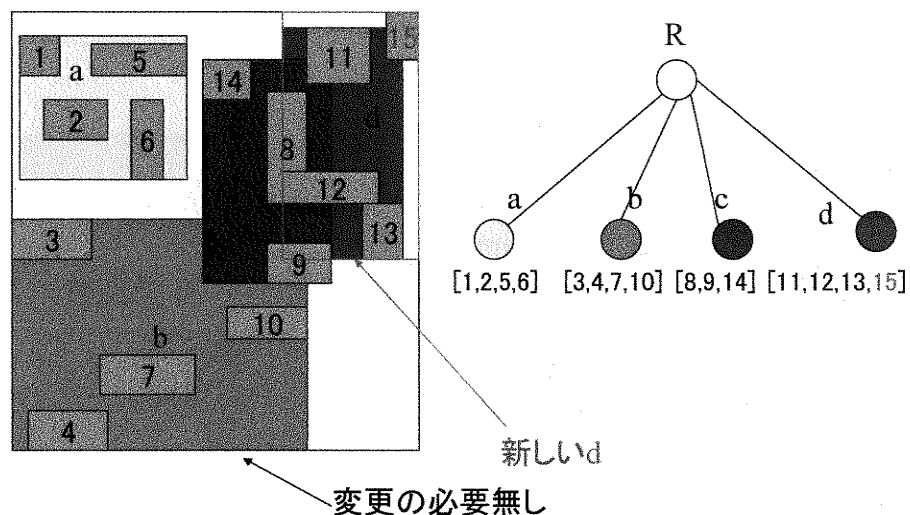
- グループの初期化
 $M+1$ 個のレコード集合 E の中で、

$$mbr(e.mbr, e'.mbr) - e.mbr - e'.mbr$$
 が最大となる e, e' を選択し、それぞれ、 v, v' にいれる。
- レコードの追加
 残りの全レコードの中で、 $d2-d1$ または $d1-d2$ が最大となるレコード e を求める。 $d2-d1$ が最大なら e を v に、 $d1-d2$ が最大なら e を v' にいれる。これを繰り返す

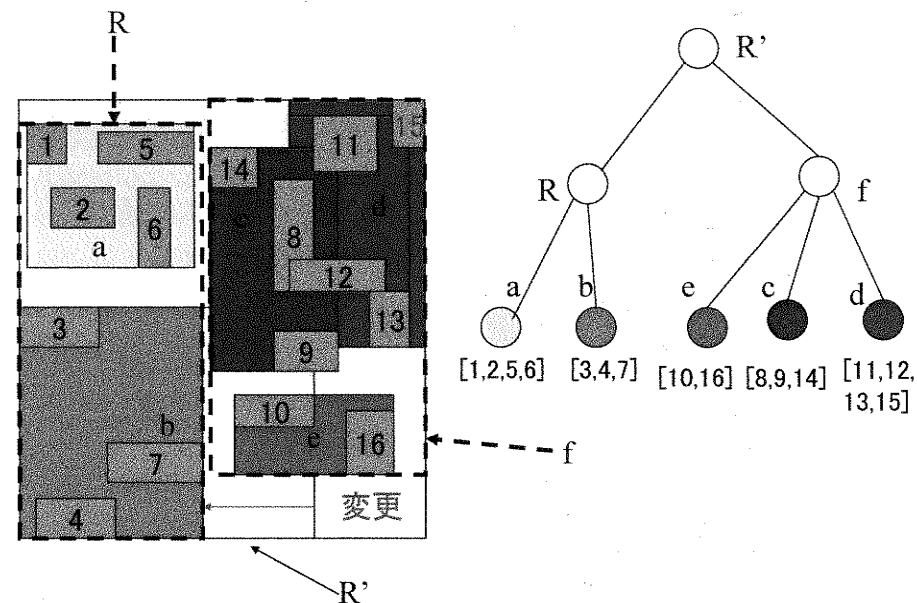
$$d1 = mbr(e.mbr, v.mbr) - e.mbr$$

$$d2 = mbr(e.mbr, v'.mbr) - e.mbr$$
- 片方のレコード数が m 未満になりそうな場合、残りのレコードをこの節点にいれる。

R木への追加例:オブジェクト15の追加



R木への追加例:オブジェクト16の追加



R+木

R木の改良

- 点質問の場合、根節点から1つの葉節点までの探索で終了できる。

各非葉節点に対応するMBRは重なりが許されない。

R+木の各レベルにおいて、節点のMBRに重なりをもつMBRはすべてその節点に含まれる。

あるレコードは複数の葉節点に重複して記憶される。

- R木よりもサイズは大きくなる。

R+木の例

