

5K-3

三次元 R*-tree のための三次元空間データ表現の比較*

堀之口浩征[†] 黒木進[‡] 牧之内顕文[§]九州大学大学院システム情報科学研究科[¶]

1 はじめに

計算機を用いて現実世界を投影するような仮想世界を構築する等の目的のために、データベースに三次元や四次元のデータを格納しようとする事が研究されて来ている。

我々の研究室では位相幾何学の概念である単体複体を用いたアプローチ Universe [1] を提案し、これに即したデータベース構築、検索手法等について研究を進めている。

今回は領域表現に適したインデックスである R*-tree[2] を三次元に拡張し、その形状表現について比較した。

2 インデックスと R*-tree

空間データを扱うインデックスの中でも R-tree[3] とその変種では、木の各ノードに自分の表現する領域（以下ではノード領域と呼ぶ。）を持たせ、親のノード領域は自分の全ての子ノードのノード領域を包含するように作る所に特徴がある。この領域に基づく階層構造が、空間質問処理を有効に処理させる事を可能とさせる。

R-tree とその変種の中でも R*-tree は、その構築過程において、木のノード選定に工夫を施す事で、検索時の効率化を実現している。

3 三次元 R*-tree

空間データベースでは物体の占める空間的領域や空間的位置等の空間データを効率的に保持する必要がある。

我々は、現在研究室内で研究を進めている Universe を空間内物体のデータ表現として利用した。

3.1 三次元 R*-tree と Universe

単体複体の概念を用いる Universe では、従来の2次元情報と三次元や四次元の物体形状の情報をシームレスに扱う事を可能にさせる利点を持つ。

物体を Universe で表現することで、管理すべきデータの集合は単体の組合せとなる。従って、扱うべき形状も少なくなるために実現は容易となる。

また、Universe では四次元空間の物体のモデル化を可能としているが、今回はその実現の容易性から Universe の三次元的な側面のみを利用している。

3.2 三次元 R*-tree とノード領域形状

インデックスでは質問条件を満たす可能性のある“候補”を出来るだけ少なく、しかも素早く引き出す事が重要課題である。

そのため、R-tree とその変種のノード領域形状には無駄の少ない包含領域の作成と領域比較の容易さが求められる。

我々は R*-tree を三次元に拡張する場合に、ノード領域の形状を変化させる事で、三次元に適した形状について調べた。

具体的には作成や比較の容易さから以下について比較した。なお、軸垂直六面体、直角四面体については便宜的な独自名称であり、また、空間を表現する座標軸として x, y, z を用いている。

- 球
- 軸平行六面体
- 直角四面体 (図 1)

x, y, z の各座標軸に垂直な平面で囲まれた六面体領域。

物体を表す点集合の x, y, z の各座標値における最小値を同時に持つ点 (点 O とする。) を一つの頂点とし、

* A comparison of three-dimensional spatial data representation in three-dimensional R*-tree

[†]Hiroyuki Horinokuchi

[‡]Susumu Kuroki

[§]Akifumi Makinouchi

[¶]Graduate School of Information Science and Electrical Engineering, Kyushu University 6-10-1 Hakozaki, Higashi, Fukuoka 812-81, Japan

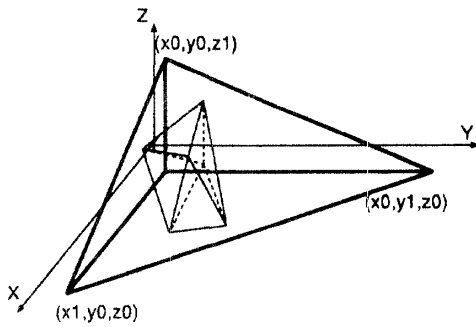


図 1: 直角四面体

点Oを共有する三面が各座標軸に垂直である性質を持つ四面体。

4 ノード領域形状の比較

今回の比較には検索時に訪問したノード数と検索の結果得られた候補数及び検索にかかる時間を用いた。インデックス比較は本質的には disk I/O を用いて論じるべきである。しかし今回の実装は検索最適化の仕組みを導入しておらず、また訪問ノード数を減らす事が間接的に disk I/O を減らす事に繋がるので、先の要素を用いた。

4.1 実験

点集合で表現される物体を多数用意したデータに対して、球、軸平行六面体、直角四面体をそのノード領域の形状とするインデックスの構築を行う。

定義域となる物体集合として、特定のノード領域形状に有利にならない形状を決める事が困難である。そこで、ノード領域形状と同じ、球、軸平行六面体、直角四面体の各々のみから成る集合を与えた。更に各集合には各々1対1で体積、位置が対応する形状を入れ、偏りが生じないように試みた。

検索では質問領域として差が出る程度に充分大きな領域を、また条件として質問領域との交差を与えた。

4.2 結果

図2では物体数を変化させ、ノード領域形状による検索ノード数の違いを対数グラフにとっている。

グラフにおける m-n(n=4,6,9) は、m が与えた検索領域の形状を、また、n がノード領域の形状を表し、4は直角四面体を、6は軸平行六面体を、9は球を意味している。

まず、検索ノード数による比較ではノード形状が軸平行六面体で有る時に他に比べて少ない特徴が見られた。4,6,9-4,9の平均に比べて6-6が46-48%程度、4-6が78-80%、9-6が83-85%と出ている。ノード形

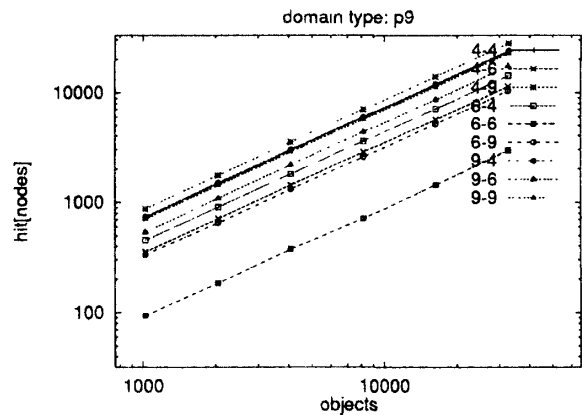


図 2: 候補出力数の推移

状による有意な差が出た事から、軸平行六面体のノードにおける無駄領域が少ないと考えられる。

また、出力候補数について比較すると、ほぼ4グループに分ける事が出来、少ない順から6-6、4-6,6-9、6-4,9-6、残りとする事が出来た。最後のグループを基準に他グループを比較するならば、13-15%、41-45%、62-68%程度であった。

軸平行六面体のノードにおける無駄領域が少ない事から、6-4,6,9の結果が良い事は説明がつく。残りの6-9,6-4については、与六面体から作られる形状が球、直角四面体の順に体積の増加が少なかったからと考えられる。

以上から、検索においては与えられた検索領域を幾つかの軸平行六面体に分割し検索を行う事が有効であると思われる。

5 おわりに

ノード形状による改良にはあまり余地が無いとの印象を得た。

我々は今回の実装及び実験により得られた知識と経験を利用した、より高次元のデータベースへの展開について現在は研究を進めている。

参考文献

- [1] 黒木進, 牧之内顕文 “単体複体の概念を用いた時空間データモデル Universe の設計” 情報処理学会研究報告, pp.221-226 1996
- [2] N. Beckmann, H.-P. Kriegel R. Schneider, B. Seeger “The R*-tree: An Efficient and Robust Access Method for Points and Rectangles” ACM SIGMOD, pp.322-331, May 1990
- [3] A. Guttman “R-trees: a dynamic index structure for spatial searching” Proc. ACM SIGMOD Int. Conf. on Management of Data, pp.47-57, 1984