韓國 컴퓨터情報學會 論文誌
第10卷 第1號, 2005. 3.

2005-3-1-1-11

다차원 인덱싱 구조에서의 k-근접객체질의 처리 방안

김 병 곤*, 오 성 균**

k-Nearest Neighbor Query processing in Multi-Dimensional Indexing Structures

Byung-Gon Kim*, Sung-Kyun Oh**

요 약

최근에 데이터베이스 응용분야에서 내용기반의 검색이 가능한 이미지 데이터와 같은 다차원 정보 처리에 대한 관심이 고조되고 있다. 따라서 다차원 데이터를 효율적으로 저장하고, 사용자가 원하는 질의 결과를 신속히 제공하는 것이 중요한 연구분야이다. 다차원의 데이터에 대한 질의는 대표적으로 영역질의(Range Query)와 최근접객체검색질의(Nearest Neighbor Query)로 나눌 수 있다. 본 논문에서는 R*-tree와 같은 다차원의 인덱싱 구조에서 효율적이고 빠른 k-근접객체검색질의를 수행하기 위한 방안을 제시한다. k-근접객체검색질의는 질의 객체로부터 가장 근접한 k개의 객체를 반환하는 것이다. 본 논문은 이를 위하여 가지치기(Pruning) 기법을 이용하여 검색 공간을 줄이는 방법을 사용하였다. 실험을 통하여 제안된 전략의 오버헤드와 이득을 보였으며, 마지막으로 가장 효율적인 전략의 사용을 제안하였다.

Abstract

Recently, query processing techniques for the multi-dimensional data like images have been widely used to perform content-based retrieval of the data. Range Quesy and Nearest neighbor query are widly used multi dimensional queries. This paper proposes the efficient pruning strategies for k-nearest neighbor query in R-tree variants indexing structures. Pruning strategy is important for the multi-dimensional indexing query processing so that search space can be reduced. We analyzed the pruning strategies and perform experiments to show overhead and the profit of the strategies. Finally, we propose best use of the strategies.

▸ Keyword : k-Nearest neighbor query, Multi-dimnesional indexing, R-tree, Pruning

• 제1저자 : 김병곤
• 접수일 : 2005.01.24, 심사완료일 : 2005.03.04
* 부천대학 e-비즈니스과 조교수 ** 서일대학 IT계열 소프트웨어전공 부교수

Ⅰ. 서 론

최근 들어 지리정보, 멀티미디어 데이터와 같은 다차원 데이터와 관련된 응용분야의 서비스에 관한 연구가 활발히 진행되고 있으며, 대량의 다차원 데이터에 대한 검색과 제공에 관한 연구가 중요한 연구과제로 부상되고 있다. 다차원 데이터를 검색하고 저장하기 위해서는 B-tree와 같은 기존의 1차원의 인덱싱 기술로는 적합하지 않으며, 다차원의 데이터를 인덱싱할 수 있는 별도의 방법들이 연구되어 왔다. 이와 관련하여 R-tree, R*tree, X-tree, TV-tree, TPR*-tree, Priority R-tree와 같은 많은 다차원 인덱싱 방법이 발표되었으며[1,2,3,4,5,6], 이를 기반으로 효율적인 질의 처리를 수행하기 위한 여러 가지 질의 처리 전략도 연구되어 왔다. 다차원의 데이터에 대한 질의는 대표적으로 영역질의(Range Query)와 최근접객체검색질의(Nearest Neighbor Query)로 나눌 수 있다. 영역질의는 질의 객체로부터 일정 거리내에 존재하는 객체를 검색하는 질의이며, 최근접객체검색질의는 질의 객체로부터 가장 근접한 혹은 가장 유사한 객체를 반환하는 것이다. 최근접객체검색질의의 변형으로 k-근접객체검색질의는 질의 객체로부터 가장 근접한 k개의 객체를 반환하는 것이다.

본 논문에서는 R*-tree와 같은 고차원의 인덱싱 구조에서 효율적이고 빠른 k-근접객체검색질의를 수행하기 위한 방안을 제시한다. 다차원 데이터 인덱스 구조 중 가장 대표적인 R*-tree는 R-tree의 삽입과 분할 알고리즘에 강제적 재삽입(*forced reinsertion*) 알고리즘을 적용하여 개선한 방법이다. R*-tree는 객체의 추가, 삭제가 용이한 동적인 특성을 지니므로 인덱스의 구조 변경에 대한 시스템의 부담이 상대적으로 적다. 그러나, R-tree 계열의 인덱스 구조는 다각형 영역간에 겹침 영역이 존재하므로, 질의 영역과 겹침이 존재하는 모든 영역 노드에 대한 방문이 요구되므로 질의의 처리 시에 많은 오버헤드를 초래한다. 또한 질의 처리 시에 질의 객체와 노드 객체들간의 유클리디언 거리를 계산하는데 많은 시간이 소요되며, 이러한 계산시간은 차원이 높고, 다각형간에 겹침 영역이 많을수록 시스템에 상당한 부담이 된다.

이와 같은 시스템 부담을 줄이기 위해서는 실제적인 객체간의 거리계산시간을 줄일수 있는 방법이 제시되어야 한다. 본 논문에서는 R-tree 계열의 인덱싱 구조에서 k-근접객체검색질의를 효율적으로 처리하기 위한 가지치기(Pruning) 기법을 제안한다. 2장에서는 관련연구에대하여 논하였으며, 3장에서는 새로운 가지치기 기법을 제안하였다. 4장에서는 기존의 방법과 제안된 방법을 실험을 통하여 비교 분석하여 제안된 방법의 효율성을 나타내었고, 5장에서 결론을 맺었다.

Ⅱ. 관련연구

앞에서 설명한 바와 같이 최근접객체검색질의는 질의 객체로부터 가장 근접한 혹은 가장 유사한 객체를 반환하는 것이다. 최근접객체검색질의의 변형으로 k-근접객체검색질의는 질의 객체로부터 가장 근접한 k개의 객체를 반환하는 것이다.

Roussopoulos, Kelly, Vincent는 최근접객체질의(NNQ) 처리시의 트리의 검색영역을 줄이기 위한 순서화(Ordering)와 가지치기(Pruning)를 위하여, MINDIST와 MINMAXDIST 정보를 사용하였다[7]. MINDIST는 MBR R로부터 질의 점 Q까지의 최소 거리를 나타내며, MINMAXDIST는 어떤 객체 O를 지니고 있는 MBR의 면으로부터 P까지의 최대거리 중에서 최소거리를 택한 것이다. MINDIST와 MINMAXDIST는 각각 최소한계와 최대한계를 의미한다. MINMAXDIST의 가장 중요한 의미는 MBR R에 있는 객체 O중에서 MINMAXDIST(P,R)보다 거리가 가까운 객체가 반드시 존재한다는 것을 보장한다는 것이다. R*-tree에서의 NNQ를 위하여 먼저 고려되어야할 것은 여러개의 후보 노드들 중에서 어느 노드를 먼저 방문하여야 하는가 이다. 이를 순서(Ordering)문제라 한다. MINDIST가 작은 노드를 먼저 방문하는 방법은 최대한 가까운 거리를 기준으로 선택하므로 낙관적 방법이라 한다. MINMAXDIST를 기준으로 순서를 정하는 것은 MINDIST에 비해 비관적인 방법이라 할 수 있다. 각 노드들을 방문할 경우 깊이 우선 검색을 하며, 두가지 방법 중 하나를 선택하는 기준은 여러 가지 요소(밀도, 분포, ..)들을 고려하여야 한다. 대부분의 경우에 MINDIST의 순서가 최적이지만 항상 그런 것은 아니다. 순서 문제와 더불어 가장 중요한 문제가 가지치기(Pruning) 문제이다. 이는 검색중 얻어낸 정보를 통하여 순회하지 않아도 되는 노

드에 대하여는 검색을 줄이는 방법이다. 위의 연구에서는 다음 세가지의 경험요소를 고려하였다.

전략 1) If MINDIST(P,M) 〉 MINMAXDIST(P,N) then MBR M 은 처리하지 않는다.

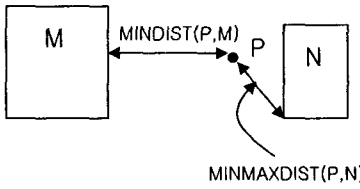


그림 1. 최근접객체검색질의 가지치기 전략 1
Fig 1. Nearest neighbor query pruning strategy 1

전략 2) If 실제거리(O,P) 〉 MINMAXDIST(P,N) then object O 는 처리하지 않는다.

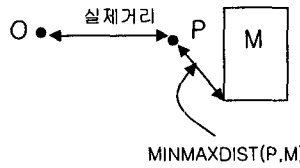


그림 2. 최근접객체검색질의 가지치기 전략 2
Fig 2. Nearest neighbor query pruning strategy 2

전략 3) If 실제거리(O,P) 〈 MINDIST(P,M) then MBR M 은 처리하지 않는다.

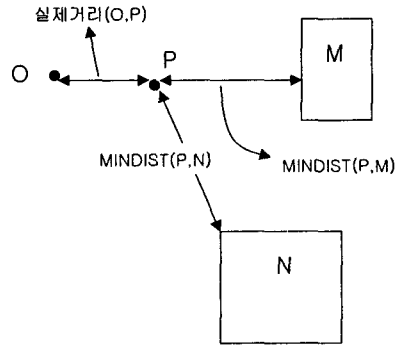


그림 3. 최근접객체검색질의 가지치기 전략 3
Fig 3. Nearest neighbor query pruning strategy 3

그러나, 위에서 제시한 가지치기 전략은 근접객체검색질의를 위한 것으로서 k-근접객체검색질의를 처리할 경우 k개의 객체를 한꺼번에 고려할 수 없다.

k-근접객체검색질의 검색을 수행하기 위한 또 하나의 연구는 Seidl과 Kriegel에 의하여 제시되었다[8]. Seidl과 Kriegel은 k-근접객체검색을 위하여 2단계의 처리전략을 제시하였다. 첫 번째 단계에서는 필터거리함수(Filter distance function)을 이용하여 후보객체의 수를 줄인다. 이때, 필터거리함수는 실제거리함수보다 적은 차원수로 객체간의 거리측정이 가능하므로, 짧은 시간 내에 후보객체들을 산출할 수 있으며, LBP(Lower Bounding Property)를 만족한다. 이렇게 산출된 후보객체들에 대하여 두 번째 단계에서는 실제거리함수를 적용하여 가장 유사한 k개의 객체를 반환하도록 하였다. 그러나 이 방법의 단점은 필터거리함수에 따라 알고리즘의 성능이 좌우될 수 있다는 것과, 단말노드에서의 필터링이 고려되지 않으므로 인해 겹침이 많은 인덱싱 환경에서는 성능저하가 발생할 수 있다.

k-근접객체검색질의를 수행하기 위하여 Bozkaya와 Ozsoyoglu는 R-treer 계열의 영역별 객체구분 방법을 사용하지 않고 객체간의 상대적인 거리를 기반으로 하는 MVP-tree구조를 발표하였다[9]. MVP-tree는 VP-tree를 보완한 거리 기반 색인 구조이다. VP-tree의 단점은 트리의 분기수가 커질 경우, 고차원 영역에서는 너무 얇은 조각이 발생한다는 것이다. 이는 검색영역이 겹치게 되고, 많은 검색영역을 검색해야하는 것을 의미한다. MVP-tree는 한 노드에서 한 개 이상의 VP를 적용하여 검색 영역에 대한 변별력을 높이도록 한 구조이다. 즉, 필터링 효과를 증가시킬 수 있다. MVP-tree방법은 질의처리시에 단말노드까지의 운행을 위하여 필요한 부가적인 거리계산의 부담을 줄였으나, 이 트리의 정적인 특성 때문에 삽입과 삭제가 빈번한 응용에서는 사용하기 힘들다는 단점이 있다.

k-근접객체검색질의를 처리하기 위한 여러 가지 시도들은 각각의 장단점을 지닌다. 각각의 단점들을 보완하는 것이 효율적인 데이터처리를 위한 방법이 된다. 본 논문에서는 [6]에서 제시한 최근접객체질의의 가지치기 기법을 보완하여 k-근접객체검색질의를 수행하는 새로운 기법을 제시한다. R-tree 계열의 인덱싱 구조는 현재까지도 여러 분야에서 사용되고 있으며, 이와 유사한 인덱싱 구조가 계속적으로 발표되고 있다. 다음 장에서는 새로운 가지치기 전략을 위하여 MAXDIST 개념을 소개하고 이를 이용한 새로운 가지치기 전략을 제시한다.

III. k-근접객체질의 처리를 위한 가지치기 전략

최근접객체검색질의를 처리하기 위한 2장에서의 전략중에서 전략1은 검색 공간을 줄이기 위한 순서화와 가지치기를 위하여 MINDIST와 MINMAXDIST정보를 사용한다. 그러나, k-최근접객체검색질의를 처리하기 위해서는 전략1에서의 MINMAXDIST 요소 대신에 MAXDIST라는 새로운 요소를 도입하여 효율적인 새로운 전략을 적용가능하게 된다. 그 이유는 전략1에서의 MINMAXDIST는 MBR R에서의 한개의 객체 O의 존재를 보장하지만, k-근접객체검색질의를 처리하기 위한 k개의 객체의 존재를 보장하지는 못한다. 즉, k-근접객체검색질의를 처리하기 위한 새로운 전략을 필요로한다. 이를 위하여 본 논문에서는 새로운 가지치기 전략을 다음과 같이 제안한다.

근접한 k 개의 객체의 존재를 보장하기 위하여, 우리는 MAXDIST라는 요소를 사용한다. MAXDIST에대한 정의와 이를 이용한 가지치기 전략은 다음과 같다. R-tree의 사각형 영역 R은 두개의 끝점인 S와 T로 표현할수 있으며, 대각선방향으로 R=(S,T)로 나타낸다. 이때, S={s1,s2,,sn} 과 T={t1,t2,,tn}에 대하여, si 〈= ti for 1〈= i 〈= n 이다.

- 정의 : 점 P에서 MBR M까지의 최대거리인 MAXDIST(P,R) 은
  MAXDIST(P,R) = $\sum |pi - ri|^2$ (1〈= i 〈= n)
  where if pi 〈 (si + ti)/2 then ri = ti
  else ri = si

- k-근접객체질의 처리를 위한 가지치기 전략 :
  if MINDIST(P,M) 〉 MAXDIST(P,N)
  then MBR M은 처리하지 않는다.

(이때, k는 노드의 엔트리 개수보다 작다고 가정한다.)

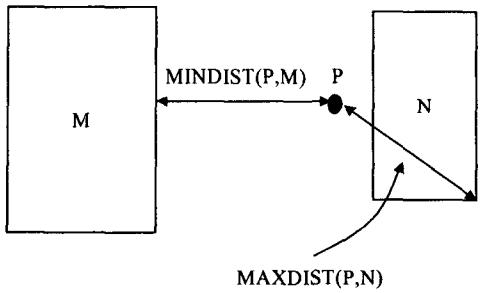


그림 4. k-근접객체검색질의 가지치기 전략
Fig 4. k-Nearest neighbor query pruning strategy

k-근접객체검색을 위하여 2장에서 제시한 3가지 전략중에서 전략1은 제시된 새로운 전략으로 대치된다. 즉 k-근접객체질의를 위한 전략으로 수정되는 것이다. 다음 장에서는 수정된 3가지 전략을 가지고 각 전략별로 성능과 효율을 측정하여 비교하였다.

IV. 실험 평가

우리는 제안된 가지치기 전략의 효율성을 보여주기 위하여 실험을 수행하였다. 가지치기 전략을 적용하기 위해서는 세가지 요소인 MINDIST, MAXDIST, MINMAXDIST를 필요로 한다.

위에서 언급한대로 세가지 요소는 n-차원의 벡터 계산에 의하여 구해지며, 이들에 대한 계산시에 CPU 시간을 소모한다. 이 세가지 요소를 이용하여 가지치기를 수행하여 질의처리 시간을 줄일수 있지만 줄일 수 있는 시간이 상대적으로 적다면 이는 오히려 시스템에 부담으로 작용할 수 있다. 그러므로, 각 전략별로 얻을 수 있는 이점과 효과를 분석하는 것이 필요하다. 따라서 우리는 각 경우에 대한 질의처리 시간을 측정하였다. 2장과 3장에서 보인대로 각 전략들은 각각 다른 요소들을 필요로 한다.

- 전략 1 : MAXDIST and MINDIST (제안 전략)
- 전략 2 : MINMAXDIST
- 전략 3 : MINDIST

실험은 100개의 샘플질의에 대하여 진행하였고, 10차원, 20차원 30차원의 200,000개의 객체를 대상으로 하였다. 실험은 R-tree의 노드당 엔트리의 개수와 k-근접객체의 k를 변화해가면서 진행되었다. 이를 통하여 MINDIST, MAXDIST, MINMAXDIST 3가지 측정요소들에 대한 계산오버헤드를 측정하고자 하였다. 따라서, 실험은 다음과 같이 네가지 경우로 나누어 진행되었다.

- 가지치기전략을 적용하지 않은 경우
- MINDIST(전략 3) 만을 적용한 경우
- MINDIST/MAXDIST (전략 1,3)를 적용한 경우
- 모든 요소를(전략 1,2,3) 사용한 경우

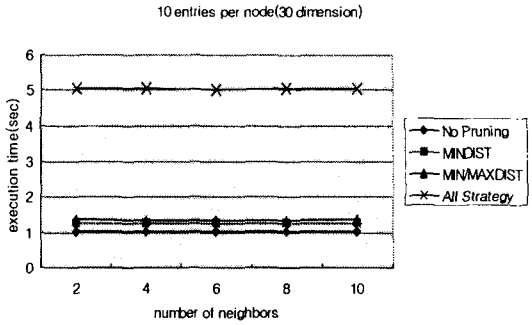
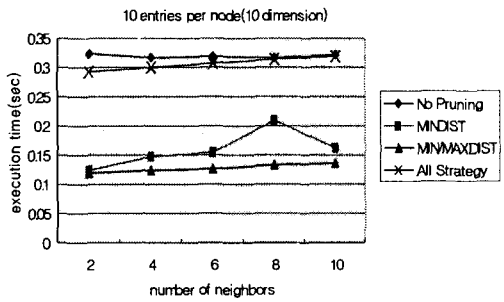


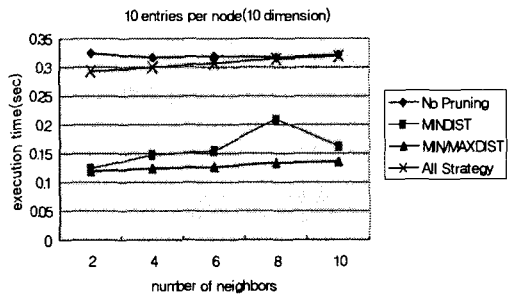
그림 5. 노드당 엔트리 10개인 경우의 질의처리시간
Fig 5. Query execution time in case of 10 entries per node

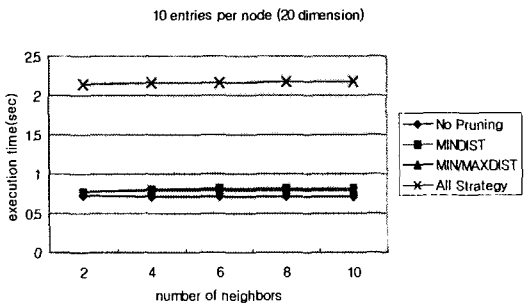
노드당 10개의 엔트리를 적용한경우(그림 5)에서는, 10차원 데이터에서는 전략1,3을 사용한 경우에 가장 좋은 결과를 보였다.

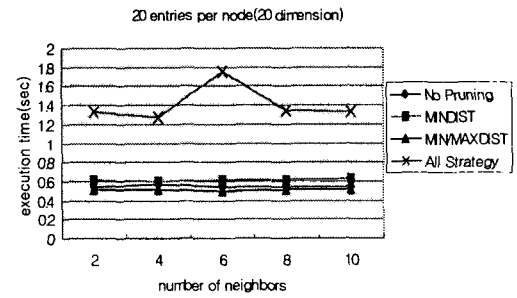
20차원과 30차원의 데이터의 경우에는 모든 요소를 사용한 경우를 제외하고는 나머지 다른 세가지 경우에는 비슷한 성능을 보였다. 결국 고차원에서는 많은 요소를 사용하여 계산하는 것이 오히려 나쁜 결과를 초래한다는 것을 알 수 있다.

노드당 20개의 엔트리를 적용한경우(그림 6)에서는, 10차원에서는 전략 1,3을 적용한 경우에 대하여 가장 좋은 결과를 보였으며, 20차원과 30차원에 대하여는, 10개의 엔트리의 경우와 같이 모든 요소를 적용한 경우가 가장 나쁜 결과를 보였다. 특히 30차원의 경우에는 가지치기를 수행하지 않은 경우에 가장 좋은 결과를 보인 것으로 보아 각 요소들을 계산하기 위한 객체간의 거리계산 오버헤드가 상당히 많은 것을 알수 있다.









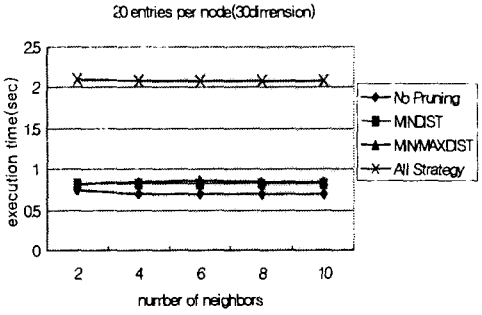
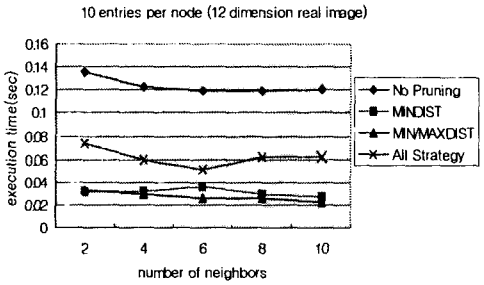
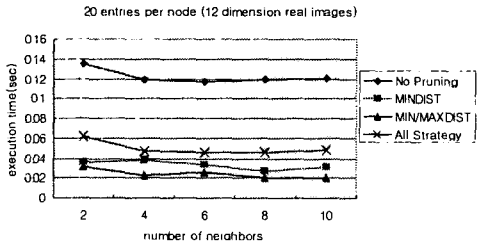


그림 6. 노드당 엔트리20개인 경우의 질의처리시간
Fig 6. Query execution time in case of 20 entries per node

다음으로, 우리는 인터넷에서 수집한 50,000개의 자연이미지에 대하여 같은 실험을 실시하였다. R-*tree 인덱싱구조를 이용하여 12차원의 데이터를 구성하였다. 그림 7에 나타난바와 같이 전략 1,3을 적용한 경우에 가장 좋은 결과를 보여주었다. 이는 앞에서 실시한 인공 데이터에 대한 실험결과와 동일한 결과를 보여주었다.

결론적으로 3가지 가지치기 전략중에서 1,3번 전략이 질의 처리 효율을 높이는 것으로 나타났으며, 전략 2는 역할을 하지 않는 것으로 나타났다.





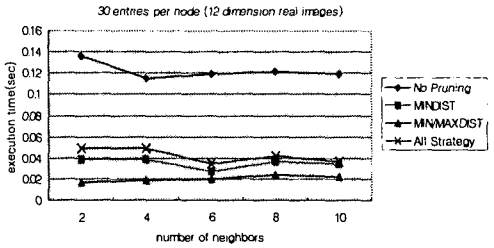


그림 7. 이미지데이터 경우의 질의처리시간
Fig 7. Query execution time in case of image data

V. 결론

본 연구에서는 k-근접객체질의를 처리하기 위한 가지치기기법을 제안하고 각 제안 전략별로 그 효용성을 실험으로 보여주었다. 가지치기 전략은 3가지로 나누어 제시되었고, 각 전략을 적용하기 위하여 세가지 요소인 MINDIST, MAXDIST, MINMAXDIST를 소개하였다. 이 세가지 요소는 가지치기에 사용되지만 이를 계산하는데 많은 오버헤드가 발생한다. 본 논문은 이 세가지 요소의 실제적인 오버헤드를 실험으로 측정하여 실제로 효용성이 있는 전략을 제시하였다.

결론적으로 MINDIST, MAXDIST의 두가지 요소를 사용하여 전략 1,3을 적용한 경우에 가장 좋은 성능을 보였다. 그러나, R-tree 계열의 인덱싱 구조의 특성상 20차원 이상의 고차원에서는 좋은 성능을 보이지 않았다. 그러므로 본 논문에서 제시하는 가지치기 기법은 10차원 이하의 응용데이터를 구현한 경우에 효용성이 높을 것으로 보인다. 논문에서 제안된 기법들은 다차원 데이터베이스 혹은 멀티미디어 데이터베이스 환경에서의 시스템구축[10]에 응용될 수 있다.

참고문헌

[1] Antonin Guttman, "R-Trees: A Dynamic Index Structure for Spatial Searching", Proceedings of the ACM SIGMOD, pages 47-57, 1984.

[2] Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider, and Bernhard Seeger, "The R*-Tree: An Efficient and Robust Access Method for Points and Rectangles", Proceedings of the ACM SIGMOD, pages 322-331, 1990.

[3] Stefan Berchtold, Daniel A. Keim, and Hans Peter Kriegel, "The X-Tree: An Index Structure for High-Dimensional Data", Proceedings of the VLDB, pages 28-39, 1996.

[4] King-Ip Lin, H. V. Jagadish, and Christos Faloutsos, "The TV-tree - An Index Structure for High-Dimensional Data", VLDB Journal, Vol. 3(4), pages 517-542, 1994.

[5] Yufei Tao, Dimitris Papadias, Jimeng Sun, "The TPR*-Tree: An Optimized Spatio-TemporalAccess Method for Predictive Queries", Proceedings of the VLDB, pages 790-801, 2003.

[6] Lars Arge, Mark de Berg, Herman J. Haverkort, Ke Yi, "The Priority R-tree : A Practically Efficient and Worst-Case Optimal R-tree", Proceedings of the ACM SIGMOD, pages 347-358, 2004.

[7] Nick Roussopoulos, Stephen Kelley, and Frederick Vincent, "Nearest Neighbor Queries", Proceedings of the ACM SIGMOD, pages 71-79, 1995.

[8] Thomas Seidl and Hans-Peter Kriegel, "Optimal Multi-Step k-Nearest Neighbor Search", Proceedings of the ACM SIGMOD Conference, pages 154-165, 1998.

[9] Tolga Bozkaya and Meral Ozsoyoglu, "Distance-Based Indexing for High-Dimensional Metric Spaces", Proceedings of the ACM SIGMOD Conference, pages 357-368, 1997.

[10] 이현창, "멀티미디어 데이터베이스 환경에서 시각화된 사용자 정의 스키마 통합", 컴퓨터정보학회 논문지 9권2호, 2004.9

저 자 소 개

김 병 곤

1990년 홍익대학교 공과대학 전자계산학과 졸업(이학사)
1992년 홍익대학교 공과대학 전자계산학과 대학원 졸업(이학석사)
2001년 홍익대학교 공과대학 전자계산학과 대학원 졸업(이학박사)
2001년 ~ 현재 부천대학 e-비즈니스과 조교수
〈관심분야〉 다차원 인덱싱, 시멘틱 웹

오 성 균

1981년 홍익대학교 이공대학 전자계산학과 졸업(이학사)
1984년 연세대학교 산업대학원 전자계산학과 졸업(공학석사)
1999년 홍익대학교 이공대학 전자계산학과 졸업(이학박사)
1987년~현재 서일대학 IT계열 소프트웨어전공 부교수
〈관심 분야〉 능동 데이터베이스, XML 모델링, 소프트웨어 공학