



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2011-0052301
(43) 공개일자 2011년05월18일

(51) Int. Cl.

H04N 5/93 (2006.01)

(21) 출원번호 10-2009-0109284

(22) 출원일자 2009년11월12일

심사청구일자 2009년11월12일

(71) 출원인

고려대학교 산학협력단

서울 성북구 안암동5가 1

(72) 발명자

정연돈

서울특별시 중구 황학동 롯데캐슬 베네치아
105-812

정하림

서울특별시 동대문구 휘경동 롯데 낙천대 아파트
101동 810호

(뒷면에 계속)

(74) 대리인

조영현, 나승택

전체 청구항 수 : 총 7 항

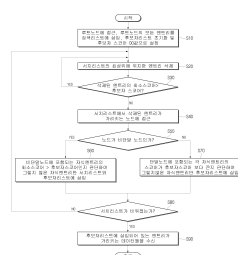
(54) 무선 데이터 방송 스트림에서 위치 기반 상위 - k 질의 처리 방법

(57) 요약

본 발명은 무선 데이터 방송 스트림에서 위치 기반 상위 - k 질의 처리 방법에 관한 것으로, 특히 무선 데이터 방송 시스템에서 데이터의 공간 속성뿐만 아니라 비공간 속성을 활용하여 이동 사용자가 선호하는 데이터를 효과적으로 검색할 수 있게 하는 새로운 위치기반 질의인 위치기반 상위-k 질의 처리 방법에 관한 것이다.

본 발명인 무선 데이터 방송 스트림에서 위치 기반 상위 - k 질의 처리 방법을 이루는 구성수단은, 단말 노드의 자식 엔트리는 자신이 가리키는 데이터의 비공간 속성값을 추가적으로 저장하고, 비단말 노드의 자식 엔트리는 자신의 엔트리에 포함되는 모든 데이터들의 비공간 속성값들에 대한 최소값 및 최대값으로 이루어진 구간값을 추가적으로 저장하는 R-tree 구조의 데이터가 순차적으로 전송되는 무선 데이터 방송 스트림에서 위치 기반 상위 - k 질의 처리 방법에 있어서, 공간 조건과 비공간 조건으로 구성된 위치 기반 상위-k 질의가 주어지면, 방송 스트림상의 루트 노드에 접근하여 루트 노드의 모든 엔트리를 검색리스트(search list)에 삽입하고, 후보자리스트를 초기화하며 후보자스코어를 무한대값으로 설정하는 제1 단계, 상기 서치리스트의 최상위에 위치한 엔트리를 삭제하고, 상기 서치리스트에서 삭제된 엔트리의 최소스코어가 상기 후보자스코어보다 큰지 판단하는 제2 단계, 상기 판단 결과, 상기 서치리스트에서 삭제된 엔트리의 최소스코어가 상기 후보자스코어보다 큰 경우에는 상기 제2 단계를 수행하고, 상기 서치리스트에서 삭제된 엔트리의 최소스코어가 상기 후보자스코어보다 크지 않는 경우에는 상기 서치리스트에서 삭제된 엔트리가 가리키는 노드에 접근하여, 이 노드가 비단말 노드인지 단말 노드인지 판단하는 제3 단계, 상기 판단 결과, 상기 노드가 비단말 노드인 경우에는 상기 비단말 노드에 포함되는 각 자식 엔트리에 대하여, 각 자식 엔트리의 최소스코어가 후보자스코어보다 큰지 판단한 후, 최소스코어가 후보자스코어보다 크지 않는 자식 엔트리만 서치리스트 및 후보자리스트에 삽입하고, 상기 노드가 단말 노드인 경우에는 데이터를 가리키는 상기 단말 노드에 포함되는 각 자식 엔트리에 대하여, 각 자식 엔트리의 데이터 스코어가 후보자스코어보다 큰지 판단한 후, 데이터 스코어가 후보자스코어보다 크지 않는 자식 엔트리만 후보자리스트에 삽입하는 제4 단계, 상기 서치리스트가 비워질 때까지 상기 제2 단계 내지 상기 제4 단계를 반복 수행한 후, 후보자리스트에 삽입되어 있는 엔트리가 가리키는 데이터들을 수신하여 결과를 도출하는 제5 단계를 포함하여 이루어진 것을 특징으로 한다.

대표도 - 도6



(72) 발명자

조병구

충청남도 서천군 화양면 옥포리 326번지

조용현

서울 동작구 대방동 381-52 대연빌라 302호

이 발명을 지원한 국가연구개발사업

과제고유번호 KRF-2008-313-D00856

부처명 한국학술진흥재단

연구관리전문기관

연구사업명 기초과학 『하반기_단독』

연구과제명 Wireless Information Push 시스템에서 공간/비공간 혼합질의를 위한 색인 및 질의처리
기법에 관한 연구

기여율

주관기관 고려대학교

연구기간 2008년 11월 01일 ~ 2009년 10월 31일

특허청구의 범위

청구항 1

단말 노드의 자식 엔트리는 자신이 가리키는 데이터의 비공간 속성값을 추가적으로 저장하고, 비단말 노드의 자식 엔트리는 자신의 엔트리에 포함되는 모든 데이터들의 비공간 속성값들에 대한 최소값 및 최대값으로 이루어진 구간값을 추가적으로 저장하는 R-tree 구조의 데이터가 순차적으로 전송되는 무선 데이터 방송 스트림에서 위치 기반 상위-k 질의 처리 방법에 있어서,

공간 조건과 비공간 조건으로 구성된 위치 기반 상위-k 질의가 주어지면, 방송 스트림상의 루트 노드에 접근하여 루트 노드의 모든 엔트리를 검색리스트(search list)에 삽입하고, 후보자리스트를 초기화하며 후보자스코어를 무한대값으로 설정하는 제1 단계;

상기 서치리스트의 최상위에 위치한 엔트리를 삭제하고, 상기 서치리스트에서 삭제된 엔트리의 최소스코어가 상기 후보자스코어보다 큰지 판단하는 제2 단계;

상기 판단 결과, 상기 서치리스트에서 삭제된 엔트리의 최소스코어가 상기 후보자스코어보다 큰 경우에는 상기 제2 단계를 수행하고, 상기 서치리스트에서 삭제된 엔트리의 최소스코어가 상기 후보자스코어보다 크지 않는 경우에는 상기 서치리스트에서 삭제된 엔트리가 가리키는 노드에 접근하여, 이 노드가 비단말 노드인지 단말 노드인지 판단하는 제3 단계;

상기 판단 결과, 상기 노드가 비단말 노드인 경우에는 상기 비단말 노드에 포함되는 각 자식 엔트리에 대하여, 각 자식 엔트리의 최소스코어가 후보자스코어보다 큰지 판단한 후, 최소스코어가 후보자스코어보다 크지 않는 자식 엔트리만 서치리스트 및 후보자리스트에 삽입하고, 상기 노드가 단말 노드인 경우에는 데이터를 가리키는 상기 단말 노드에 포함되는 각 자식 엔트리에 대하여, 각 자식 엔트리의 데이터 스코어가 후보자스코어보다 큰지 판단한 후, 데이터 스코어가 후보자스코어보다 크지 않는 자식 엔트리만 후보자리스트에 삽입하는 제4 단계;

상기 서치리스트가 비워질 때까지 상기 제2 단계 내지 상기 제4 단계를 반복 수행한 후, 후보자리스트에 삽입되어 있는 엔트리가 가리키는 데이터들을 수신하여 결과를 도출하는 제5 단계를 포함하여 이루어진 것을 특징으로 하는 무선 데이터 방송 스트림에서 위치 기반 상위-k 질의 처리 방법.

청구항 2

청구항 1에 있어서,

상기 비단말 노드의 자식 엔트리는 상기 구간값 대신에 비트벡터를 추가적으로 저장하되, 상기 비트벡터는 k개의 비트로 이루어진 비트열이고, 각 비트는 k개의 클러스터에 각각 일대일 대응되고, 상기 클러스터 각각은 상기 비공간 속성값들을 독립적인 구간으로 나누기 위한 클러스터 최소값과 클러스터 최대값으로 이루어진 클러스터 구간값을 가지며, 상기 비단말 노드의 자식 엔트리가 저장하고 있는 비트벡터는 상기 비단말 노드의 자식 엔트리에 포함되는 데이터들의 비공간 속성값들을 포함하는 클러스터 구간값을 가지는 클러스터에 대응되는 비트만 "1"의 값을 가지는 것을 특징으로 하는 무선 데이터 방송 스트림에서 위치 기반 상위-k 질의 처리 방법.

청구항 3

청구항 1 또는 청구항 2에 있어서,

상기 서치리스트에서 삭제된 엔트리가 가리키는 노드에 접근한 후에, 상기 후보자리스트에 상기 서치리스트에서 삭제된 엔트리가 삽입되어 있는 경우에는 상기 후보자리스트에서 상기 서치리스트에서 삭제된 엔트리를 삭제하는 과정을 수행하는 것을 특징으로 하는 무선 데이터 방송 스트림에서 위치 기반 상위-k 질의 처리 방법.

청구항 4

청구항 1 또는 청구항 2에 있어서,

상기 노드가 비단말 노드인 경우에 최소스코어가 후보자스코어보다 크지 않는 자식 엔트리만 서치리스트에 삽입하는 단계는 엔트리의 전송 순서대로 정렬하여 삽입하는 단계인 것을 특징으로 하는 무선 데이터 방송 스트림에서 위치 기반 상위 -k 질의 처리 방법.

청구항 5

청구항 1 또는 청구항 2에 있어서,

상기 후보자리스트는 k개까지의 엔트리를 저장하되, 상기 후보자리스트에 저장된 엔트리들은 각 엔트리들의 최대스코어 또는 데이터의 스코어가 오름차순으로 정렬될 수 있도록 저장되고, 상기 후보자스코어는 상기 후보자리스트에 k개 미만의 엔트리가 저장되어 있는 경우에는 무한대값을 유지하고, k개 엔트리가 저장되어 있는 경우에는 상기 후보자리스트에 k번째 저장되어 있는 엔트리의 최대스코어 또는 데이터의 스코어인 것을 특징으로 하는 무선 데이터 방송 스트림에서 위치 기반 상위 -k 질의 처리 방법.

청구항 6

청구항 5에 있어서,

상기 노드가 비단말 노드인 경우에 최소스코어가 후보자스코어보다 크지 않는 자식 엔트리만 후보자리스트에 삽입하는 단계는 현재 후보자리스트에 저장되어 있는 엔트리 개수가 k개 미만인 경우에는 엔트리들의 최대스코어 또는 데이터의 스코어가 오름차순으로 정렬될 수 있도록 삽입하고, k개인 경우에는 상기 자식 엔트리의 최대스코어가 상기 후보자스코어보다 작은 경우에만 상기 후보자리스트의 k번째 엔트리를 삭제한 후, 엔트리들의 최대스코어 또는 데이터의 스코어가 오름차순으로 정렬될 수 있도록 삽입되는 것을 특징으로 하는 무선 데이터 방송 스트림에서 위치 기반 상위 -k 질의 처리 방법.

청구항 7

청구항 5에 있어서,

상기 노드가 단말 노드인 경우에 데이터 스코어가 후보자스코어보다 크지 않는 자식 엔트리만 후보자리스트에 삽입하는 단계는 현재 후보자리스트에 저장되어 있는 엔트리 개수가 k개 미만인 경우에는 엔트리들의 최대스코어 또는 데이터 스코어가 오름차순으로 정렬될 수 있도록 삽입하고, k개인 경우에는 상기 후보자리스트의 k번째 엔트리를 삭제한 후, 엔트리들의 최대스코어 또는 데이터 스코어가 오름차순으로 정렬될 수 있도록 삽입되는 것을 특징으로 하는 무선 데이터 방송 스트림에서 위치 기반 상위 -k 질의 처리 방법.

명세서

발명의 상세한 설명

기술분야

[0001] 본 발명은 무선 데이터 방송 스트림에서 위치 기반 상위 -k 질의 처리 방법에 관한 것으로, 특히 무선 데이터 방송 시스템에서 데이터의 공간 속성뿐만 아니라 비공간 속성을 활용하여 이동 사용자가 선호하는 데이터를 효과적으로 검색할 수 있게 하는 새로운 위치기반 질의인 위치기반 상위-k 질의 처리 방법에 관한 것이다.

배경기술

[0002] 무선 통신 및 모바일 정보 기술의 발달에 힘입어 위치기반 서비스(location based services)에 대한 연구가 국내/외적으로 활발히 진행되고 있다. 한편, 무선 데이터 방송(wireless data broadcasting)은 서버가 불특정 다수의 이동사용자에게 방송 채널(broadcasting channel)을 통해 데이터를 일방적으로 푸시(push)하고, 각각의 사용자는 휴대용 컴퓨터를 사용하여 방송 채널을 통해 자신이 원하는 데이터를 선별적으로 수신하는 통신 방식

이다.

- [0003] 위치 기반 서비스를 효과적으로 지원하기 위해서는 공간범위 질의 (spatial range queries) 혹은 k-최근접 이웃 질의(k-nearest neighbor queries) 등과 같이 사용자의 현재 위치를 기반으로 하여 유용한 정보를 검색하는 위치기반 질의(location based queries)의 효율적인 수행이 필수적이다.
- [0004] 하지만 기존의 위치기반 질의는 사용자의 위치와 데이터 사이의 공간적인 거리만을 고려하므로, 데이터의 공간 속성뿐만 아니라 다양한 비공간 속성을 함께 고려하여 사용자가 선호하는 정보를 제공하는 실질적인 위치기반 서비스를 지원하기에는 한계를 지닌다.
- [0005] 무선 데이터 방송 시스템에서 서버는 불특정 다수의 이동 사용자에게 방송 채널을 통해 일방적으로 데이터를 푸시 한다. 사용자는 별도의 송신 작업 없이 방송 채널을 통해 자신이 원하는 데이터를 수신한다.
- [0006] 따라서 각각의 사용자와 서버 사이에 별도의 통신 채널이 필요한 개별 통신 방식의 시스템에 반해, 하나의 방송 채널을 모든 사용자들이 공유하는 무선 데이터 방송 시스템은 한정된 무선 네트워크 대역폭을 효율적으로 활용할 수 있다. 또한 송신 작업에 소요되는 에너지를 절약할 수 있으므로 한정된 에너지 자원을 가진 휴대용 컴퓨터의 에너지 효율성을 향상시킬 수 있다.
- [0007] 무선 데이터 방송 시스템에서 사용자는 신속하게 원하는 데이터를 수신할 수 있어야 한다. 또한 사용자는 배터리로 동작하는 휴대용 단말기를 사용하므로 에너지 효율성을 고려한 데이터 수신이 요구된다. 무선 데이터 방송 시스템에서 널리 사용되는 성능 측정의 척도는 다음과 같다[T.Imielinski, S.Viswanathan and B.R.Bardrinath, "Data on Air: Organization and Access," IEEE Trans. Knowledge and Data Engineering, 9(3), 1997.].
- [0008] 1) 접근 시간: 휴대용 컴퓨터가 방송 채널을 듣기 시작한 시점으로부터 사용자가 원하는 데이터를 모두 수신한 시점까지의 소요 시간으로서 사용자의 평균 검색 시간을 가늠할 수 있다.
- [0009] 2) 튜닝 시간: 접근 시간 동안 실제로 방송 채널을 들은 시간으로서 휴대용 컴퓨터의 에너지 효율성을 가늠할 수 있다.
- [0010] 사용자가 적은 에너지 소모로 자신이 원하는 데이터만을 선택적으로 수신할 수 있도록 하기 위해 방송 스트림 상에서 데이터의 위치, 즉 데이터의 전송 시간 정보를 색인 구조로 구축한 후 데이터와 함께 방송 스트림에 배치하는 색인 인터리빙 (index interleaving) 기법이 널리 사용되고 있다.
- [0011] 색인 인터리빙 기법을 통해 사용자는 원하는 데이터가 전송되는 시점을 계산하고, 에너지 보존 상태인 대기 상태 (doze mode)로 기다리다가 필요로 하는 데이터가 전송되는 시점에서만 활성 상태 (active mode)로 전환하여 방송 채널을 듣고 데이터를 수신할 수 있다.
- [0012] 따라서 튜닝 시간을 줄일 수 있다. 하지만 색인 정보의 추가로 인하여 방송 스트림의 길이가 증가하게 되어 사용자의 평균 접근 시간이 늘어나게 된다. 따라서 색인 인터리빙 기법은 사용자의 접근 시간과 튜닝 시간을 동시에 고려해야 한다.
- [0013] [T.Imielinski, S.Viswanathan and B.R.Bardrinath, "Data on Air: Organization and Access," IEEE Trans. Knowledge and Data Engineering, 9(3), 1997.]에서 제안한 (1,m) 색인 인터리빙 기법은 방송 스트림에 색인을 m번 중복/배치하는 가장 기본적인 방법이다. 도 1은 (1,m) 색인 인터리빙 기법의 예를 보여준다.
- [0014] 방송 스트림을 구성하는 가장 작은 논리적 단위는 버킷(bucket)이다. 데이터를 포함하는 버킷을 데이터 버킷, 색인을 포함하는 버킷을 색인 버킷이라 부르며 데이터 버킷과 색인 버킷의 용량은 동일하다고 가정한다. 데이터 버킷과 색인 버킷의 용량이 동일하고 방송 채널의 대역폭은 고정되어 있으므로 버킷의 수는 쉽게 시간 값으로 변환할 수 있다. 따라서 접근 시간과 튜닝 시간은 버킷의 수로 측정할 수 있다.
- [0015] R-tree[A. Guttman, "R-trees: A Dynamic Index Structure for Spatial Searching," In SIGMOD, 1984.]는 B+-tree의 다차원 확장으로서 공간 데이터 혹은 다차원 데이터의 효율적인 질의 처리를 위해 가장 널리 이용되고 있다. 도 2는 본 발명의 실시예를 설명하기 위한 샘플 지역 레스토랑 데이터를 보여주고, 도 3은 도 2의 레스토랑 데이터를 색인하는 R-tree를 보여준다.
- [0016] 공간 범위 질의와 k-최근접 이웃 질의는 가장 일반적인 위치기반 질의이다. 도 3과 같이 R-tree는 공간적으로 서로 인접한 데이터들의 위치를 mbr(minimum bounding rectangle)로 근사화하여 계층화된 트리 구조를 구축하므로, 공간 범위 질의의 수행에 매우 효과적이다. 루트를 시작으로 주어진 검색 범위와 겹치는 노트

- [0017] 만을 접근 및 검사한다.
- [0018] R-tree 상에서 k-최근접 이웃 질의의 효율적인 수행에 초점을 맞춘 많은 알고리즘들이 제안되었다[G.R. Hjaltason, H. Samet, "Distance Browsing in Spatial Databases," ACM Trans. Database Systems, 24(2), 1999.와 K.L. Cheung, A. Fu, "Enhanced Nearest Neighbour Search on the R-tree," SIGMOD Record, 27(3), 1998.와 N. Roussopoulos, S. Kelley, and F.Vincent, "Nearest Neighbor Queries," In SIGMOD, 1995. 참조].
- [0019] 이들은 대부분 [N. Roussopoulos, S. Kelley, and F.Vincent, "Nearest Neighbor Queries," In SIGMOD, 1995.]에서 제안한 거리척도인 mindist와 minmaxdist를 활용한 분기 한정(branch-and-bound) 탐색 방식을 따른다.
- [0020] 질의점 q.P와 R-tree 비단말 노드(non-leaf node)의 엔트리 E가 주어졌을 때, mindist(q.P, E.mbr)는 E.mbr이 이 거리보다 가까운 데이터를 포함하지 않음을 보장한다. 한편, minmaxdist(q.P, E.mbr)는 E.mbr이 이 거리보다 가까운 데이터를 적어도 1개 이상 포함함을 보장한다.
- [0021] 도 3에서는 간략함을 위해 mindist(q.P, E₆.mbr)와 minmaxdist(q.P, E₆.mbr)만을 보이고 있다.
- [0022] 분기 한정 탐색 방식은 백트래킹 (backtracking)이 자주 발생한다. 이는 순차적으로 색인구조 및 데이터가 전송되는 방송 스트림의 특성으로 인해, 무선 데이터 방송 시스템에서 사용자의 접근 시간을 상당히 증가시킬 수 있다. 예를 들어, 도 4와 같이 탐색 알고리즘이 N₂를 접근한 후 N₁을 접근하려면 다음에 방송되는 R-tree를 기다려야 한다.
- [0023] 이러한 방송 스트림의 순차적 특성을 반영하여 사용자의 접근시간을 줄이기 위해, [B. Gedik, A. Singh, and L. Liu, "Energy Efficient Exact kNN Search in Wireless Broadcast Environments," In ACM-GIS, 2004.와 B. Zheng, W.-C. Lee, and D. Lee, "Search k Nearest Neighbors on Air," In MDM, 2003.]에서 제안한 알고리즘들은 mindist와 minmaxdist를 활용한 순차적 탐색 방식을 따른다.
- [0024] 이러한 공간 데이터를 색인하는 기존의 R-tree는 데이터의 공간 속성만을 다루므로 위치기반 상위-k 질의 처리에는 적절하지가 못하다.

발명의 내용

해결 하고자하는 과제

- [0025] 본 발명은 상기와 같은 종래기술의 문제점을 해결하기 위하여 창안된 것으로, 특히 데이터의 공간 속성뿐만 아니라 비공간 속성을 활용하여 사용자가 선호하는 정보를 효과적으로 검색할 수 있게 하는 무선 데이터 방송 스트림에서 위치 기반 상위 -k 질의 처리 방법을 제공하는 것을 그 목적으로 한다.

과제 해결수단

- [0026] 상기와 같은 과제를 해결하기 위하여 제안된 본 발명인 무선 데이터 방송 스트림에서 위치 기반 상위 -k 질의 처리 방법을 이루는 구성수단은, 단말 노드의 자식 엔트리는 자신이 가리키는 데이터의 비공간 속성값을 추가적으로 저장하고, 비단말 노드의 자식 엔트리는 자신의 엔트리에 포함되는 모든 데이터들의 비공간 속성값들에 대한 최소값 및 최대값으로 이루어진 구간값을 추가적으로 저장하는 R-tree 구조의 데이터가 순차적으로 전송되는 무선 데이터 방송 스트림에서 위치 기반 상위 -k 질의 처리 방법에 있어서, 공간 조건과 비공간 조건으로 구성된 위치 기반 상위-k 질의가 주어지면, 방송 스트림상의 루트 노드에 접근하여 루트 노드의 모든 엔트리를 검색 리스트(search list)에 삽입하고, 후보자리스트를 초기화하며 후보자스코어를 무한대값으로 설정하는 제1 단계, 상기 서치리스트의 최상위에 위치한 엔트리를 삭제하고, 상기 서치리스트에서 삭제된 엔트리의 최소스코어가 상기 후보자스코어보다 큰지 판단하는 제2 단계, 상기 판단 결과, 상기 서치리스트에서 삭제된 엔트리의 최소스코어가 상기 후보자스코어보다 큰 경우에는 상기 제2 단계를 수행하고, 상기 서치리스트에서 삭제된 엔트리의 최소스코어가 상기 후보자스코어보다 크지 않는 경우에는 상기 서치리스트에서 삭제된 엔트리가 가리키는 노드에 접근하여, 이 노드가 비단말 노드인지 단말 노드인지 판단하는 제3 단계, 상기 판단 결과, 상기 노드가 비단말

노드인 경우에는 상기 비단말 노드에 포함되는 각 자식 엔트리에 대하여, 각 자식 엔트리의 최소스코어가 후보 자스코어보다 큰지 판단한 후, 최소스코어가 후보자스코어보다 크지 않는 자식 엔트리만 서치리스트 및 후보자 리스트에 삽입하고, 상기 노드가 단말 노드인 경우에는 데이터를 가리키는 상기 단말 노드에 포함되는 각 자식 엔트리에 대하여, 각 자식 엔트리의 데이터 스코어가 후보자스코어보다 큰지 판단한 후, 데이터 스코어가 후보 자스코어보다 크지 않는 자식 엔트리만 후보자리스트에 삽입하는 제4 단계, 상기 서치리스트가 비워질 때까지 상기 제2 단계 내지 상기 제4 단계를 반복 수행한 후, 후보자리스트에 삽입되어 있는 엔트리가 가리키는 데이터 들을 수신하여 결과를 도출하는 제5 단계를 포함하여 이루어진 것을 특징으로 한다.

[0027] 또 다른 본 발명인 무선 데이터 방송 스트림에서 위치 기반 상위 -k 질의 처리 방법을 이루는 구성수단은, 상기 비단말 노드의 자식 엔트리는 상기 구간값 대신에 비트벡터를 추가적으로 저장하되, 상기 비트벡터는 k개의 비트로 이루어진 비트열이고, 각 비트는 k개의 클러스터에 각각 일대일 대응되고, 상기 클러스터 각각은 상기 비 공간 속성값들을 독립적인 구간으로 나누기 위한 클러스터 최소값과 클러스터 최대값으로 이루어진 클러스터 구간값을 가지며, 상기 비단말 노드의 자식 엔트리가 저장하고 있는 비트벡터는 상기 비단말 노드의 자식 엔트리에 포함되는 데이터들의 비공간 속성값들을 포함하는 클러스터 구간값을 가지는 클러스터에 대응되는 비트만 "1"의 값을 가지는 것을 특징으로 한다.

[0028] 여기서, 상기 서치리스트에서 삭제된 엔트리가 가리키는 노드에 접근한 후에, 상기 후보자리스트에 상기 서치리스트에서 삭제된 엔트리가 삽입되어 있는 경우에는 상기 후보자리스트에서 상기 서치리스트에서 삭제된 엔트리를 삭제하는 과정을 수행하는 것을 특징으로 한다.

[0029] 여기서, 상기 노드가 비단말 노드인 경우에 최소스코어가 후보자스코어보다 크지 않는 자식 엔트리만 서치리스트에 삽입하는 단계는 엔트리의 전송 순서대로 정렬하여 삽입하는 단계인 것을 특징으로 한다.

[0030] 여기서, 상기 후보자리스트는 k개까지의 엔트리를 저장하되, 상기 후보자리스트에 저장된 엔트리들은 각 엔트리들의 최대스코어 또는 데이터의 스코어가 오름차순으로 정렬될 수 있도록 저장되고, 상기 후보자스코어는 상기 후보자리스트에 k개 미만의 엔트리가 저장되어 있는 경우에는 무한대값을 유지하고, k개 엔트리가 저장되어 있는 경우에는 상기 후보자리스트에 k번째 저장되어 있는 엔트리의 최대스코어 또는 데이터의 스코어인 것을 특징으로 한다.

[0031] 여기서, 상기 노드가 비단말 노드인 경우에 최소스코어가 후보자스코어보다 크지 않는 자식 엔트리만 후보자리스트에 삽입하는 단계는 현재 후보자리스트에 저장되어 있는 엔트리 개수가 k개 미만인 경우에는 엔트리들의 최대스코어 또는 데이터의 스코어가 오름차순으로 정렬될 수 있도록 삽입하고, k개인 경우에는 상기 자식 엔트리의 최대스코어가 상기 후보자스코어보다 작은 경우에만 상기 후보자리스트의 k번째 엔트리를 삭제한 후, 엔트리들의 최대스코어 또는 데이터의 스코어가 오름차순으로 정렬될 수 있도록 삽입되는 것을 특징으로 한다.

[0032] 여기서, 상기 노드가 단말 노드인 경우에 데이터 스코어가 후보자스코어보다 크지 않는 자식 엔트리만 후보자리스트에 삽입하는 단계는 현재 후보자리스트에 저장되어 있는 엔트리 개수가 k개 미만인 경우에는 엔트리들의 최대스코어 또는 데이터 스코어가 오름차순으로 정렬될 수 있도록 삽입하고, k개인 경우에는 상기 후보자리스트의 k번째 엔트리를 삭제한 후, 엔트리들의 최대스코어 또는 데이터 스코어가 오름차순으로 정렬될 수 있도록 삽입되는 것을 특징으로 한다.

효 과

[0033] 상기와 같은 과제 및 해결수단을 가지는 본 발명인 무선 데이터 방송 스트림에서 위치 기반 상위 -k 질의 처리 방법에 의하면, 비단말 노드의 자식 엔트리들이 자신의 엔트리에 포함된 모든 데이터들의 비공간 속성값들에 대한 최소값 및 최대값으로 이루어진 경계값 또는 작은 사이즈로 부호화하기 위한 비트 벡터를 추가적으로 저장하고 있기 때문에, 무선 데이터 방송 스트림에서 질의 처리를 위한 튜닝시간 및 접근 시간 성능을 더욱 향상시키는 장점이 있다.

발명의 실시를 위한 구체적인 내용

[0034] 이하, 첨부된 도면을 참조하여 상기와 같은 과제, 해결수단 및 효과를 가지는 본 발명인 무선 데이터 방송 스트림에서 위치 기반 상위 -k 질의 처리 방법에 관한 바람직한 실시예를 상세하게 설명한다.

- [0035] 본 발명인 무선 데이터 방송 스트림에서 위치 기반 상위-k 질의 처리 방법에 관한 구체적인 실시예를 설명하기 전에, 본 발명에 적용되는 정의와 데이터 색인 구조에 대하여 설명한다.
- [0036] 데이터의 집합 $D=\{d_1, d_2, \dots, d_{|D|}\}$ 에서 각각의 데이터 d 는 공간 속성 $P=(x,y)$ 와 비공간 속성의 집합 $A=\{a_1, a_2, \dots, a_n\}$ 를 포함하며, $(d.P, d.A)$ 로 표기한다. 즉, $d.P=(d.x, d.y)$ 는 d 의 위치이고, $d.A=\{d.A_1, d.A_2, \dots, d.A_n\}$ 는 d 의 비공간 속성 값들의 집합이다.
- [0037] D 에 대한 위치기반 상위-k 질의 q 는 $(q.P, q.V)$ 로 표기한다. 여기서 $q.P=(q.x, q.y)$ 는 사용자가 명시한 자신의 위치이고, $q.V=\{q.V_1, q.V_2, \dots, q.V_n\}$ 는 사용자가 A 에 대한 조건으로 명시한 값들의 집합이다.
- [0038] 위치기반 질의 q 의 기본 목적은 q 가 명시한 공간/비공간 조건에 가장 근접한 상위 k 개의 데이터를 찾는 것이며, q 에 대한 데이터 d 의 근접 정도를 의미하는 스코어(score)는 다음과 같이 (수식 1)에 의하여 판별한다.
- [0039]
$$\text{score}(q,d) = \text{dist}(q.P, d.P) + \text{dist}(q.V, d.A) \quad (\text{수식 1})$$
- [0040] 여기서, $\text{dist}(q.P, d.P) = \sqrt{(q.x - d.x)^2 + (q.y - d.y)^2}$ 는 q 와 d 의 공간 거리이며, $\text{dist}(q.V, d.A) = \sum_{i=1}^n |q.V_i - d.A_i|$ 는 q 와 d 의 비공간 거리이다. 또한 $\text{dist}(q.P, d.P)$ 와 $\text{dist}(q.V, d.A)$ 는 같은 범위의 값을 가지도록 정규화된다.
- [0041] 여기서 상기 $\text{dist}(q.P, d.P)$ 와 $\text{dist}(q.V, d.A)$ 에는 가중치를 부가할 수 있다. 따라서 부가되는 가중치에 따라 상기 스코어 $\text{score}(q,d)$ 의 값을 달라질 수 있다. 또한, 상기 데이터는 복수개의 비공간 속성을 가질 수 있다.
- [0042] 그런데, 이하에서는 본 발명의 설명을 용이하게 하기 위하여 상기 공간 조건과 비공간 조건은 동일하게 중요하게 고려되는 것으로 한다. 즉, 상기 $\text{dist}(q.P, d.P)$ 와 $\text{dist}(q.V, d.A)$ 에 가중치 부가 없이 스코어 $\text{score}(q,d)$ 를 판별한다. 또한, 상기 데이터는 1개의 비공간 속성만을 포함하는 것으로 가정한다. 즉, $\text{dist}(q.V, d.A) = |q.V - d.A|$ 가 성립한다.
- [0043] 본 발명의 제1 실시예에 따른 무선 데이터 방송 스트림에서 위치 기반 상위-k 질의 처리 방법은 공간 데이터를 색인하는 기존의 R-tree를 확장한 색인 구조를 따른다. 즉, 비단말 노드의 엔트리에 데이터의 비공간 속성 값들에 대한 집계정보를 추가적으로 저장하는 aR-tree(aggregate R-tree)[D. Papadias, P. Kalnis, J. Zhang, and Y. Tao, "Efficient OLAP Operations in Spatial Data Warehouses," In SSTD, 2001.와 I. Lazaridis and S. Mehrotra, "Progressive Approximate Aggregate Queries with a Multi-Resolution Tree Structure," In SIGMOD, 2001. 참조]를 이용한다.
- [0044] 특히, aR-tree의 비단말 노드의 엔트리 E 가 주어졌을 때, $E.mbr$ 에 공간적으로 포함되는 모든 데이터의 집합을 subtree(E)라고 하자. 이때, 본 발명에서 적용되는 R-tree를 확장한 aR-tree는 다음과 같은 특성을 지닌다.
- [0045] (1) 단말 노드의 자식 엔트리(단말 엔트리)는 자신이 가리키는 데이터의 비공간 속성값을 추가적으로 저장한다. 즉, 데이터의 공간 속성값뿐만 아니라, 도 2에서 보여준 바와 같이, 레스토랑의 평균 가격과 같은 비공간 속성값을 추가적으로 저장한다.
- [0046] (2) 비단말 노드의 자식 엔트리(비단말 엔트리) E 는 subtree(E)에 포함된 데이터의 비공간 속성값들에 대한 최소값 및 최대값으로 이루어진 구간값 $ivl = [\underline{ivl}, \overline{ivl}]$ 을 추가적으로 저장한다. 즉, 공간 속성값 이외의 구간값인 비공간 속성값을 추가적으로 저장한다.
- [0047] 도 5는 도 2에서 보여준 레스토랑 데이터를 색인하는 aR-tree를 보여준다. 예를 들어, 레스토랑 r_7 의 메뉴 평균 가격은 \$10이다. 따라서 r_7 을 가리키는 단말 엔트리는 r_7 의 비공간 속성값인 10을 추가적으로 저장한다. 한편, $E_2.mbr$ 은 레스토랑 r_5, r_6, r_7, r_8 을 공간적으로 포함하며 이들의 평균 메뉴가격은 각각 \$18, \$4, \$10, \$8이다. 따라서, 비단말 엔트리 E_2 는 subtree(E_2)에 포함된 레스토랑의 메뉴가격들에 대한 최소값 및 최대값으로 이루어진 1-차원 구간 $ivl=[4, 18]$ 을 추가적으로 저장한다.
- [0048] 상기와 같은 설명은 결국, 각 비단말 엔트리 E 가 저장하고 있는 1-차원 구간값 $E.ivl = [\underline{ivl}, \overline{ivl}]$ 이 주어졌을

때, subtree(E)에 포함된 데이터의 비공간 속성값들은 모두 $\underline{ivl} \leq d.A \leq \overline{ivl}$ 은 만족함을 알 수 있다.

[0049] 정리하면, 본 발명에 적용되는 확장된 R-tree 구조는 단말 노드의 자식 엔트리는 자신이 가리키는 데이터의 비공간 속성값을 추가적으로 저장하고, 비단말 노드의 자식 엔트리는 자신의 엔트리에 포함되는 모든 데이터들의 비공간 속성값들에 대한 최소값 및 최대값으로 이루어진 구간값을 추가적으로 저장한다.

[0050] 결국, 제1 실시예에 따른 본 발명은 단말 노드의 자식 엔트리는 자신이 가리키는 데이터의 비공간 속성값을 추가적으로 저장하고, 비단말 노드의 자식 엔트리는 자신의 엔트리에 포함되는 모든 데이터들의 비공간 속성값들에 대한 최소값 및 최대값으로 이루어진 구간값을 추가적으로 저장하는 R-tree 구조의 데이터가 순차적으로 전송되는 무선 데이터 방송 스트림에서 위치 기반 상위-k 질의 처리 방법에 관한 것이다.

[0051] 본 발명에 따른 무선 데이터 방송 스트림에서 위치 기반 상위-k 질의 처리는 사용자의 휴대용 컴퓨터 등과 같은 무선 단말기에서 실행되며, 순차적으로 aR-tree 및 데이터가 전송되는 방송 스트림의 특성을 반영하여 순차적 탐색 방식을 따른다. 또한, aR-tree는 깊이우선(depth-first) 방식으로 방송 스트림에 배치된다고 가정한다.

[0052] 상기와 같은 데이터 트리 구조가 스트림으로 전송될 때, 위치 기반 상위-k 질의 처리를 수행하는데, 이에 대한 상세한 설명 전에, 추가적으로 위치기반 상위-k 질의와 각 aR-tree(R-tree를 확장한 트리)의 비단말 엔트리 사이에 정의되는 근접 정도 척도인 최소스코어(lower score)와 최대스코어(upper score)에 대하여 정의한다.

[0053] 위치기반 상위-k 질의 q와 aR-tree의 비단말 엔트리 E가 주어졌을 때, q에 대한 E의 최소스코어(lower score)는 다음과 같이 (수식 2)에 의하여 결정된다.

[0054]
$$\text{lowerscore}(q, E) = \text{mindist}(q, P, E.mbr) + \text{mindist}(q, V, E.ivl) \quad (\text{수식 2})$$

[0055] 여기서
$$\text{mindist}(q, V, E.ivl) = \begin{cases} |q.V - \underline{ivl}|, & \text{if } q.V < \underline{ivl} \\ |q.V - \overline{ivl}|, & \text{if } q.V > \overline{ivl} \end{cases}$$
 이고, 나머지 경우(q.V가 최소값 및 최대값 사이의 값인 경우)에는 mindist(q.V, E.ivl)가 "0"의 값을 가진다.

[0056] 상기와 같은 정의를 통하여 위치기반 상위-k 질의 q와 aR-tree의 비단말 엔트리 E가 주어졌을 때, $\forall d \in \text{subtree}(E), \text{lowerscore}(q, E) \leq \text{score}(q, d)$ 임을 보장할 수 있다. 이에 대하여 증명하면 다음과 같다.

[0057] 질의 q가 명시한 조건에 대한 각각의 데이터 $d \in \text{subtree}(E)$ 의 스코어(score)는 (수식 1)에 나타난 바와 같이, dist(q.P, d.P)와 dist(q.V, d.V)의 합으로 판별되며, q에 대한 E의 최소스코어(lower score)는 (수식 2)에 나타난 바와 같이, mindist(q.P, E.mbr)와 mindist(q.V, E.ivl)의 합으로 결정된다.

[0058] 또한 [N. Roussopoulos, S. Kelley, and F.Vincent, "Nearest Neighbor Queries," In SIGMOD, 1995.]에서 이미 $\forall d \in \text{subtree}(E), \text{mindist}(q, P, E.mbr) \leq \text{dist}(q, P, d.P)$ 임을 증명하였기 때문에, $\forall d \in \text{subtree}(E), \text{mindist}(q, V, E.ivl) \leq \text{dist}(q, V, d.V)$ 임을 밝히면 된다. 이를 증명하기 위하여 세 가지 경우로 나누어서 설명한다.

[0059] 1) 경우 1: $q.V < \underline{ivl}$ 이면, $\text{mindist}(q, V, E.ivl)$ 는 $|q.V - \underline{ivl}|$ 이다. 또한 $\forall d \in \text{subtree}(E), \underline{ivl} \leq d.A \leq \overline{ivl}$ 이므로 $\forall d \in \text{subtree}(E), |q.V - \underline{ivl}| \leq |q.V - d.A|$ 이 성립한다. 따라서, $\text{mindist}(q, V, E.ivl) \leq \text{dist}(q, V, d.V)$ 임을 보장할 수 있다.

[0060] 2) 경우 2 : $q.V > \overline{ivl}$ 인 경우에는 상기 경우 1의 증명과 동일한 방법으로 보장할 수 있다.

[0061] 3) 경우 3 : $\underline{ivl} \leq q.V \leq \overline{ivl}$ 이면, $\text{mindist}(q, V, E.ivl) = 0$ 이므로 항상 $\text{mindist}(q, V, E.ivl) \leq \text{dist}(q, V, d.V)$ 임을 보장할 수 있다.

[0062] 위치기반 상위-k 질의 q와 aR-tree의 비단말 엔트리 E가 주어졌을 때, q에 대한 E의 최대스코어(upper score)는 다음과 같이 (수식 3)에 의하여 결정된다.

- [0063] $upperscore(q, E) = \minmaxdist(q, P, E.mbr) + \maxdist(q, V, E.ivl)$ (수식 3)
- [0064] 여기서, $\maxdist(q, V, E.ivl) = \max(|q.V - \underline{ivl}|, |q.V - \overline{ivl}|)$ 이다.
- [0065] 상기와 같은 정의를 통하여, 위치 기반 상위-k 질의 q와 aR-tree의 비단말 엔트리 E가 주어졌을 때, $\exists d \in subtree(E) : score(q, d) \leq upperscore(q, E)$ 임을 보장할 수 있다.
- [0066] 이를 증명하면, 최대스코어(upperscore)는 $\minmaxdist(q, P, E.mbr)$ 와 $\maxdist(q, V, E.ivl)$ 의 합으로 결정되며, [N. Roussopoulos, S. Kelley, and F.Vincent, "Nearest Neighbor Queries," In SIGMOD, 1995.]에서 이미 $subtree(E)$ 는 $dist(q, P, \hat{d}.P) \leq \minmaxdist(q, P, E.mbr)$ 임을 만족하는 데이터 $\hat{d}$ 를 포함함을 증명하였으므로, $dist(q, V, \hat{d}.A) \leq \maxdist(q, V, E.ivl)$ 임만을 증명하면 된다.
- [0067] 상술한 바와 같이, $\maxdist(q, V, E.ivl) = \max(|q.V - \underline{ivl}|, |q.V - \overline{ivl}|)$ 이고, $\hat{d} \in subtree(E)$ 이므로 $\underline{ivl} \leq \hat{d}.A \leq \overline{ivl}$ 이며, $|q.V - \hat{d}.A| \leq \max(|q.V - \underline{ivl}|, |q.V - \overline{ivl}|)$ 이 성립한다. 따라서, $dist(q, V, \hat{d}.A) \leq \maxdist(q, V, E.ivl)$ 이다.
- [0068] 이상에서 설명한 정의들은 질의 처리 과정에서 탐색 범위를 줄이기 위한 토대가 된다. 질의 q와 aR-tree의 비단말 엔트리 E가 주어졌을 때, 만약 최소스코어 $lowerscore(q, E)$ 보다 q가 명시한 조건에 더 근접한 k개의 데이터가 이미 존재한다면, 탐색 과정에서 E를 탐색 대상에서 제외시킬 수 있다. 즉, E가 가리키는 노드 N을 접근할 필요가 없다.
- [0069] 한편, $subtree(E)$ 는 최대스코어 $upperscore(q, E)$ 보다 q가 명시한 조건에 더 근접한 데이터를 적어도 한 개 이상 포함하므로, 최대스코어 $upperscore(q, E)$ 는 또 다른 aR-tree의 비단말 엔트리 $\hat{E}$ 에 대한 고려 여부의 판단 기준이 될 수 있다. 예를 들어, $k=1$ 일 경우, $upperscore(q, E) < lowerscore(q, \hat{E})$ 라면, $\hat{E}$ 는 안전하게 고려 대상에서 제외시킬 수 있다.
- [0070] 이상에서 설명한 조건들 및 정의들을 적용하여 본 발명인 무선 데이터 방송 스트림에서 위치 기반 상위-k 질의 처리 방법에 관하여 도 6을 참조하여 설명한다.
- [0071] 본 발명은 확장된 R-tree 구조(aR-tree 구조)의 데이터가 순차적으로 전송되는 무선 데이터 방송 스트림에서 위치 기반 상위-k 질의를 처리하는 방법에 관한 것이고, 상기 확장된 R-tree 구조(aR-tree 구조)는 상술한 바와 같이, 단말 노드의 자식 엔트리는 자신이 가리키는 데이터의 비공간 속성값을 추가적으로 저장하고, 비단말 노드의 자식 엔트리는 자신의 엔트리에 포함되는 모든 데이터들의 비공간 속성값들에 대한 최소값 및 최대값으로 이루어진 구간값을 추가적으로 저장하는 구조이다.
- [0072] 먼저, 공간 조건과 비공간 조건으로 구성된 위치 기반 상위-k 질의가 사용자에게 의하여 주어진다. 즉, 사용자의 위치(공간 조건)와 사용자가 원하는 주변 레스토랑의 가격(비공간 조건) 및 사용자가 찾고자 하는 사용자 위치와 사용자가 원하는 가격에 근접한 레스토랑 개수가 주어진다.
- [0073] 그러면, 방송 스트림 상의 루트 노드에 접근하여 루트 노드의 모든 엔트리를 검색리스트(search list)에 삽입하고, 후보자리스트(candidate list)를 초기화하며 후보자스코어(candidate score)를 무한대값으로 설정한다(s10).
- [0074] 여기서, 상기 검색리스트는 탐색 과정에서 상기 상위-k 질의를 만족하는지를 꼭 고려해야하는 비단말 엔트리만을 방송 스트림에 배치된 순서대로 정렬하여 저장한다. 따라서, 고려 대상으로 해야하는 비단말 엔트리들을 상기 검색리스트에 저장한 다음, 하나씩 삭제해 가면서 상기 상위-k 질의를 만족하는지를 확인한다.
- [0075] 그리고, 상기 후보자리스트는 탐색 진행 과정에서 현재까지의 임시 상위-k 데이터를 포함하는(또는 가리키는) 비단말 엔트리(또는 단말 엔트리)를 최대스코어(또는 데이터의 스코어)로 오름차순으로 정렬하여 저장한다. 사용자가 만족하는 데이터를 k개만 찾는 것으로 제안한 경우, 상기 후보자리스트에는 k개까지만 저장된다.
- [0076] 즉, 상기 후보자리스트는 최대 k개의 엔트리를 저장할 수 있고, 이때 상기 k번째 엔트리의 최대스코어(upperscore)(또는 데이터의 스코어)를 후보자스코어라고 칭한다. 구체적으로, 상기 후보자리스트에 저장되어

있는 엔트리가 k 개 미만인 경우에는 상기 후보자스코어가 처음 세팅된 무한대값을 유지하고, 상기 후보자리스트에 저장되어 있는 엔트리가 k 개인 경우에는 k 번째 저장되어 있는 엔트리의 최대스코어 또는 데이터의 스코어가 상기 후보자스코어가 된다.

- [0077] 상기와 같이, 방송 채널을 통해 aR-tree의 루트(Root) 노드에 접근하여 루트 노드의 모든 엔트리들을 서치리스트에 삽입하고, 후보자리스트를 초기화하며(비우며) 후보자스코어를 무한대값으로 설정한 후에는 다음과 같은 절차들을 상기 서치리스트가 모두 비워질 때까지 서치리스트에 저장되어 있는 엔트리들에 대하여 반복적으로 하나씩 검사한다.
- [0078] 먼저, 서치리스트의 최상위에 위치한 엔트리를 삭제하고(s20), 상기 서치리스트에서 삭제된 엔트리의 최소스코어가 상기 후보자스코어보다 큰지 판단한다(s30). 상기 서치리스트는 상술한 바와 같이, 방송 스트림을 통해 방송되는 엔트리 순서대로 정렬되어 있다. 따라서, 상기 삭제되는 최상위 엔트리는 저장되어 있는 엔트리들 중에, 가장 먼저 방송되는 엔트리이다.
- [0079] 상기 판단 결과, 상기 서치리스트에서 삭제된 엔트리의 최소스코어가 상기 후보자스코어보다 큰 경우에는 상기 서치리스트의 최상위에 위치한 엔트리를 삭제하는 단계(s20)를 수행한다. 최초 검색에서는 상기 후보자스코어가 무한대값으로 설정되어 있기 때문에 이와 같은 단계는 바로 발생하지 않는다.
- [0080] 반면, 상기 서치리스트에서 삭제된 엔트리의 최소스코어가 상기 후보자스코어보다 크지 않는 경우에는 상기 서치리스트에서 삭제된 엔트리가 가리키는 노드에 접근한다(s40). 이때, 추가적으로 상기 서치리스트에서 삭제된 엔트리가 가리키는 노드에 접근한 후에, 상기 서치리스트에서 삭제된 엔트리가 상기 후보자리스트에 삽입(저장)되어 있는지 판단한다. 그 결과, 상기 후보자리스트에 상기 서치리스트에서 삭제된 엔트리가 삽입되어 있는 경우에는 상기 후보자리스트에서 상기 서치리스트에서 삭제된 엔트리를 삭제한다.
- [0081] 상기와 같이 상기 서치리스트에서 삭제된 엔트리가 가리키는 노드에 접근한 후에는, 상기 서치리스트에서 삭제된 엔트리가 가리키는 노드가 비단말 노드인지 단말 노드인지 판단한다(s50).
- [0082] 상기와 같이 상기 서치리스트에서 삭제된 엔트리가 가리키는 노드가 비단말 노드인지 단말 노드인지 판단한 결과, 상기 노드가 비단말 노드인 경우에는 상기 비단말 노드에 포함되는 각 자식 엔트리에 대하여 각 자식 엔트리의 최소스코어가 후보자스코어보다 큰지 판단한다(s60).
- [0083] 상기 판단 결과, 상기 비단말 노드에 포함되는 자식 엔트리의 최소스코어가 후보자스코어보다 큰 경우에는 k -질의 만족하는 데이터를 상기 자식 엔트리가 포함하고 있지 않은 경우이기 때문에, 상기 자식 엔트리에 대해서는 더 이상 고려하지 않고 무시한 후, 다른 자식 엔트리의 최소스코어가 후보자스코어보다 큰지 판단한다.
- [0084] 반대로, 상기 판단 결과, 상기 비단말 노드에 포함되는 자식 엔트리의 최소스코어가 후보자스코어보다 크지 않는 경우에는, 상기 자식 엔트리를 서치리스트 및 후보자리스트에 삽입한다(s60). 즉, 각 자식 엔트리의 최소스코어가 후보자스코어보다 큰지 판단한 후, 최소스코어가 후보자스코어보다 크지 않는 자식 엔트리만 서치리스트 및 후보자리스트에 삽입한다(s60).
- [0085] 여기서, 상기 서치리스트에 자식 엔트리를 삽입하는 과정은 저장되는 엔트리들이 방송 스트림 상에서 전송되는 전송 순서대로 정렬하여 삽입된다. 예를 들어, 상기 자식 엔트리가 서치리스트에 삽입(저장)되기 전에 다른 엔트리들이 상기 서치리스트에 저장되어 있고, 상기 자식 엔트리가 상기 이미 저장되어 있는 다른 엔트리들보다 먼저 방송되는 엔트리인 경우에는, 상기 자식 엔트리는 상기 서치리스트의 첫번째 공간에 저장된다.
- [0086] 그리고, 상기 후보자리스트는 상술한 바와 같이, k 개까지의 엔트리를 저장하되, 상기 후보자리스트에 저장된 엔트리들은 각 엔트리들의 최대스코어 또는 데이터의 스코어가 오름차순으로 정렬될 수 있도록 저장되고, 상기 후보자스코어는 상기 후보자리스트에 k 개 미만의 엔트리가 저장되어 있는 경우에는 무한대값을 유지하고, k 개 엔트리가 저장되어 있는 경우에는 상기 후보자리스트에 k 번째 저장되어 있는 엔트리의 최대스코어 또는 데이터의 스코어가 된다.
- [0087] 구체적으로 상기 자식 엔트리를 후보자리스트에 삽입하는 과정을 살펴보면, 상기와 같이 상기 노드가 비단말 노드인 경우에 최소스코어가 후보자스코어보다 크지 않는 자식 엔트리만 후보자리스트에 삽입하는 단계(s60)는 현재 후보자리스트에 저장되어 있는 엔트리 개수가 k 개 미만인 경우에는 저장되는 엔트리들의 최대스코어 또는 데이터의 스코어가 오름차순으로 정렬될 수 있도록 삽입한다.
- [0088] 만약, 상기 현재 후보자리스트에 저장되어 있는 엔트리 개수가 k 개인 경우에는 상기 자식 엔트리의 최대스코어가 상기 후보자스코어보다 큰지 판단한 후, 상기 자식 엔트리의 최대스코어가 상기 후보자스코어보다 작은 경우

에만 상기 후보자리스트의 k번째 엔트리를 삭제한 후, 상기 자식 엔트리를 상기 후보자리스트에 저장되는 엔트리들의 최대스코어 또는 데이터의 스코어가 오름차순으로 정렬될 수 있도록 삽입한다.

- [0089] 지금까지는 서치리스트에 삭제된 엔트리가 가리키는 노드가 비단말 노드인 경우에 대하여 설명하였고, 다음은 서치리스트에서 삭제된 엔트리가 가리키는 노드가 단말 노드인 경우에 대하여 설명한다.
- [0090] 상기 서치리스트에서 삭제된 엔트리가 가리키는 노드가 단말 노드인 경우에는 데이터를 가리키는 상기 단말 노드에 포함되는 각 자식 엔트리에 대하여, 각 자식 엔트리의 데이터 스코어가 후보자스코어보다 큰지 판단한 후, 데이터 스코어가 후보자스코어보다 크지 않는 자식 엔트리만을 후보자리스트에 삽입한다(s70).
- [0091] 즉, 각 자식 엔트리의 데이터 스코어가 후보자스코어보다 큰지 판단하여, 자식 엔트리의 데이터 스코어가 후보자스코어보다 큰 경우에는 k-질의를 만족하는 데이터가 아닌 것으로 판단하여 무시하고, 자식 엔트리의 데이터 스코어가 후보자스코어보다 크지 않는 경우에만 해당 자식 엔트리만 후보자리스트에 삽입한다.
- [0092] 여기서, 상기 후보자리스트는 상술한 바와 같이, k개까지의 엔트리를 저장하되, 상기 후보자리스트에 저장된 엔트리들은 각 엔트리들의 최대스코어 또는 데이터의 스코어가 오름차순으로 정렬될 수 있도록 저장되고, 상기 후보자스코어는 상기 후보자리스트에 k개 미만의 엔트리가 저장되어 있는 경우에는 무한대값을 유지하고, k개 엔트리가 저장되어 있는 경우에는 상기 후보자리스트에 k번째 저장되어 있는 엔트리의 최대스코어 또는 데이터의 스코어가 된다.
- [0093] 구체적으로 상기 자식 엔트리를 후보자리스트에 삽입하는 과정을 살펴보면, 상기과 같이 상기 노드가 단말 노드인 경우에 데이터 스코어가 후보자스코어보다 크지 않는 자식 엔트리만 후보자리스트에 삽입하는 단계(s70)는 현재 후보자리스트에 저장되어 있는 엔트리 개수가 k개 미만인 경우에는 엔트리들의 최대스코어 또는 데이터 스코어가 오름차순으로 정렬될 수 있도록 삽입한다.
- [0094] 만약, 상기 현재 후보자리스트에 저장되어 있는 엔트리 개수가 k개인 경우에는 상기 후보자리스트의 k번째 엔트리를 삭제한 후, 엔트리들의 최대스코어 또는 데이터 스코어가 오름차순으로 정렬될 수 있도록 삽입한다.
- [0095] 상술한 바와 같이, 서치리스트에서 삭제된 엔트리가 가리키는 노드가 비단말 노드인지 단말 노드인지 판단하여 (s50), 이의 결과에 따라 진행된 상기 단계 s60 또는 s70을 수행한 후에는 서치리스트가 비워졌는지 확인한다 (s80).
- [0096] 상기 확인 결과, 서치리스트가 비워진 상태가 아니고, 엔트리가 저장되어 있는 경우에는 상기 서치리스트가 비워질 때까지 상술한 절차들(s20 ~ s70)을 반복 수행한다. 즉, 서치리스트가 비워진 상태가 아닌 경우에는 현재 서치리스트에 저장된 최상위에 위치한 엔트리를 삭제하고(s20), 삭제된 엔트리의 최소스코어가 후보자스코어보다 큰지 판단하고(s30), 그런 경우에는 상기 단계 s20을 수행하고, 그렇지 않은 경우에는 서치리스트에 삭제된 엔트리가 가리키는 노드에 접근하여, 이 노드가 비단말 노드인지 단말 노드인지를 판단하며, 그 판단 결과에 따라 단계 s60 또는 s70을 수행한다.
- [0097] 반면, 서치리스트가 비워졌는지를 확인한 결과, 비워진 경우에는 현재 후보자리스트에 삽입되어 있는 엔트리가 가리키는 데이터들을 수신하여 결과를 도출한다(s90). 즉, 더 이상 고려할 엔트리가 없고, k-질의를 만족시키는 데이터들을 모두 찾은 상태이므로, 후보자리스트에 삽입되어 있는 엔트리가 가리키는 데이터들을 수신한다.
- [0098] 이하, 상술한 무선 데이터 방송 스트림에서 위치 기반 상위-k 질의 처리 방법에 관한 구체적인 예를 설명한다.
- [0099] 구체적인 설명을 위하여, 도 3에 도시된 바와 같은 색인 구조를 기본적으로 따르고, 사용자가 q.P에 위치하고 주변에 복수개의 레스토랑이 위치하고 있다고 가정하자. 그리고, 도 4에 도시된 방송 스트림으로 순차 방송되며, 도 5에 도시된 바와 같이, 각 엔트리가 해당 데이터를 저장하고 있다고 가정한다. 또한, 주변 레스토랑의 평균 가격은 도 2에 도시된 바와 같다고 가정한다.
- [0100] 더 구체적으로, 루트(Root)는 노드 N_1 을 가리키는 비단말 엔트리 E_1 과 노드 N_2 를 가리키는 비단말 엔트리 E_2 를 포함하고, N_1 은 노드 N_3 을 가리키는 비단말 엔트리 E_3 과 노드 N_4 를 가리키는 비단말 엔트리 E_4 를 포함하고, N_2 는 노드 N_5 를 가리키는 비단말 엔트리 E_5 와 노드 N_6 를 가리키는 비단말 엔트리 E_6 을 포함한다. 이와 같은 구조로, 부모 노드는 자식 노드를 가리키는 특정 비단말 엔트리와 다른 자식 노드를 가리키는 다른 특정 비단말 엔트리를 포함하고 있다.

[0101] 또한, 노드 N_6 은 r_7 을 가리키는 단말 엔트리와 r_8 을 가리키는 단말 엔트리를 포함하고 있다. 이와 같은 구조로, 해당 노드는 특정 데이터를 가리키는 단말 엔트리와 다른 특정 데이터를 가리키는 다른 단말 엔트리를 포함하고 있다.

[0102] 상기와 같은 구조를 방송 스트림으로 방송할 때, 클라이언트의 위치는 도 3에 도시된 바와 같이, (10, 8)이고, 클라이언트는 k(2) 질의를 만족하는 레스토랑 2개를 찾으며, 희망 가격은 10\$ 가격을 원하고 있다. 또한 클라이언트가 루트(Root)부터 데이터를 읽었다고 가정하고, 깊이 우선 방식으로 스트림에 배치한 방송 데이터를 읽는다고 가정한다.

[0103] 이 때, 데이터의 스코어(score(q, d))와 이를 결정짓는 dist(q.P, d.P)와 dist(q.V, d.A)의 계산 결과는 다음 표 1과 같다.

dist(q.P, d.P)	dist(q.V, d.A)	score(q, d)
dist(q.P, r1)=7.07	dist(q.V, r1)=6	score(q, r1)=13.07
dist(q.P, r2)=7.07	dist(q.V, r2)=2	score(q, r1)=9.07
dist(q.P, r3)=2	dist(q.V, r3)=10	score(q, r1)=12
dist(q.P, r4)=9.84	dist(q.V, r4)=5	score(q, r1)=14.84
dist(q.P, r5)=7.07	dist(q.V, r5)=8	score(q, r1)=15.07
dist(q.P, r6)=7.21	dist(q.V, r6)=6	score(q, r1)=13.21
dist(q.P, r7)=3.6	dist(q.V, r7)=0	score(q, r1)=3.6
dist(q.P, r8)=5.38	dist(q.V, r8)=2	score(q, r1)=7.38

[0105] - 표 1 -

[0106] 또한 위치 기반 상위-k 질의 q와 aR-tree의 비단말 엔트리 E가 주어졌을 때, q에 대한 E의 최소스코어(lower score(q, E))와 이를 결정하는 mindist(q.P, E.mbr)와 mindist(q.V, E.ivl)의 계산 결과는 다음 표 2와 같다.

mindist(q.P, E.mbr)	mindist(q.V, E.ivl)	lowerscore(q, E)
mindist(q.P, E1.mbr)=2	mindist(q.V, E1.ivl)=0	lowerscore(q, E1)=2
mindist(q.P, E2.mbr)=0	mindist(q.V, E2.ivl)=0	lowerscore(q, E2)=0
mindist(q.P, E3.mbr)=5.09	mindist(q.V, E3.ivl)=2	lowerscore(q, E3)=7.09
mindist(q.P, E4.mbr)=2	mindist(q.V, E4.ivl)=0	lowerscore(q, E4)=2
mindist(q.P, E5.mbr)=5.65	mindist(q.V, E5.ivl)=0	lowerscore(q, E5)=5.65
mindist(q.P, E6.mbr)=3	mindist(q.V, E6.ivl)=0	lowerscore(q, E6)=3

[0108] - 표 2 -

[0109] 또한 위치 기반 상위-k 질의 q와 aR-tree의 비단말 엔트리 E가 주어졌을 때, q에 대한 E의 최대스코어(upper score(q, E))와 이를 결정하는 minmaxdist(q.P, E.mbr)와 maxdist(q.V, E.ivl)의 계산 결과는 다음 표 3과 같다.

minmaxdist(q.P, E.mbr)	maxdist(q.V, E.ivl)	upperscore(q, E)
minmaxdist(q.P, E1.mbr)=5.38	maxdist(q.V, E1.ivl)=10	upperscore(q, E1)=15.38
minmaxdist(q.P, E2.mbr)=6.32	maxdist(q.V, E2.ivl)=8	upperscore(q, E2)=14.32
minmaxdist(q.P, E3.mbr)=7.07	maxdist(q.V, E3.ivl)=6	upperscore(q, E3)=13.07
minmaxdist(q.P, E4.mbr)=3.6	maxdist(q.V, E4.ivl)=10	upperscore(q, E4)=13.6
minmaxdist(q.P, E5.mbr)=7.07	maxdist(q.V, E5.ivl)=8	upperscore(q, E5)=15.07
minmaxdist(q.P, E6.mbr)=3.6	maxdist(q.V, E6.ivl)=2	upperscore(q, E6)=5.6

[0111] - 표 3 -

[0112] 상기와 같은 상황에서, 공간 조건과 비공간 조건으로 구성된 위치 기반 상위-k 질의가 주어지면(k는 2), 먼저, 루트(Root)를 읽고 루트(Root)가 포함하는 엔트리 E1, E2를 서치리스트(search list)에 삽입(저장)한다. 이때, 서치리스트(search list)에는 {E1, E2}를 저장하고 있고, 후보자리스트(candidate list)에는 { } (초기화 상태,

즉 비워 있는 상태)을 저장하고 있으며, 후보자스코어(candidatescore)는 무한대값이다.

- [0113] 다음, 서치리스트의 최상위에 위치한 E1을 서치리스트(searchlist)에서 삭제하고, 상기 서치리스트에서 삭제한 엔트리 E1의 최소스코어가 상기 후보자스코어보다 큰지 판단한다. 상기 삭제한 엔트리 E1의 최소스코어는 2(표 2 참조)이고 상기 후보자스코어는 무한대값이기 때문에, 상기 삭제한 엔트리 E1의 최소스코어는 상기 후보자스코어보다 크지 않다.
- [0114] 따라서, 상기 서치리스트에서 삭제된 엔트리 E1가 가리키는 노드 N1에 접근하여 읽어들인다. 이때, 서치리스트(search list)에는 {E2}을 저장하고 있고(E1이 삭제된 후이므로), 후보자리스트(candidatelist)는 { }을(초기화 상태, 즉 비워 있는 상태)저장하고 있으며, 후보자스코어(candidatescore)는 무한대값이다.
- [0115] 다음, 상기 서치리스트에서 삭제된 엔트리 E1이 가리키는 노드 N1이 단말 노드인지, 비단말 노드인지 판단한다. 상기 노드 N1은 비단말 노드이다. 따라서, 상기 비단말 노드 N1에 포함되는 각 자식 엔트리(E3, E4)에 대하여, 각 자식 엔트리의 최소스코어가 후보자스코어보다 큰지 판단한다.
- [0116] 먼저, 상기 비단말 노드 N1에 포함되는 자식 엔트리 E3에 대하여 판단하면, 상기 자식 엔트리 E3의 최소스코어는 7.09이고(표 2 참조), 후보자스코어는 무한대값이다. 따라서, 상기 자식 엔트리 E3의 최소스코어가 후보자스코어보다 크지 않고, 결과적으로 상기 자식 엔트리 E3를 서치리스트 및 후보자리스트에 삽입한다.
- [0117] 마찬가지로, 상기 비단말 노드 N1에 포함되는 자식 엔트리 E4에 대하여 판단하면, 상기 자식 엔트리 E4의 최소스코어는 2이고(표 2 참조), 후보자스코어는 무한대값이다. 따라서, 상기 자식 엔트리 E4의 최소스코어가 후보자스코어보다 크지 않고, 결과적으로 상기 자식 엔트리 E4를 서치리스트 및 후보자리스트에 삽입한다.
- [0118] 지금까지, 상기 서치리스트(search list)에는 {E3, E4, E2}을 저장하고 있고, 후보자리스트(candidatelist)에는 {E3, E4}을 저장하고 있으며, 후보자스코어(candidatescore)는 13.6의 값이다.
- [0119] 여기서, 상기 서치리스트에 저장되는 엔트리들은 엔트리의 전송 순서대로 정렬하여 삽입 저장된다. 즉, 스트림을 통하여 읽혀지는 순서대로 삽입 저장된다. 그리고, 상기 후보자리스트에는 저장되는 엔트리의 최대스코어가 오름차순으로 정렬될 수 있도록 삽입 저장된다. 즉, 상기 엔트리 E3의 최대스코어는 13.07이고, 엔트리 E4의 최대스코어는 13.6이기 때문에, 최대스코어의 오름차순으로 정렬되어 저장될 수 있도록, 상기 후보자리스트에는 {E3, E4}가 저장되어 있다.
- [0120] 또한, 상기 후보자스코어는 상기 후보자리스트의 k(클라이언트가 2로 설정함)번째 최대스코어인 13.6의 값이 된다. 이때, 상기 후보자리스트에 저장되어 있는 엔트리의 개수가 k(2)개 미만(0개 또는 1개)인 경우에는 상기 후보자스코어가 무한대값을 유지하나, 상기 후보자리스트에 저장되어 있는 엔트리의 개수가 k개(2개)이기 때문에, 후보자리스트에 저장되어 있는 k번째 엔트리의 최대스코어가 후보자스코어가 된다.
- [0121] 다음, 상기 서치리스트가 비워진 상태가 아니기 때문에, 다시 상기 서치리스트에 저장되어 있는 최상위에 위치한 엔트리 E3을 삭제하고, 상기 서치리스트에서 삭제한 엔트리 E3의 최소스코어가 상기 현재 후보자스코어보다 큰지 판단한다.
- [0122] 상기 서치리스트에서 삭제한 엔트리 E3의 최소스코어는 7.09이고, 상기 후보자스코어는 13.6이므로, 상기 서치리스트에서 삭제한 엔트리 E3의 최소스코어가 후보자스코어보다 크지 않다. 따라서, 상기 서치리스트에서 삭제한 엔트리 E3가 가리키는 노드 N3에 접근하여 읽어들인다.
- [0123] 여기서, 상기 서치리스트에서 삭제된 엔트리 E3가 가리키는 노드에 접근하여 읽어들인 후에는, 상기 후보자리스트에 상기 서치리스트에서 삭제한 엔트리 E3가 삽입 저장되어 있는지 판단한다. 판단 결과, 상기 후보자리스트에 상기 서치리스트에서 삭제된 엔트리 E3가 삽입되어 있기 때문에, 상기 후보자리스트에서 상기 서치리스트에서 삭제된 엔트리 E3를 삭제하는 과정을 수행한다.
- [0124] 이 때, 상기 서치리스트(search list)에는 {E4, E2}을 저장하고 있고(E3가 삭제된 후이기 때문에), 후보자리스트(candidatelist)에는 {E4}을 저장하고 있으며(E3가 삭제된 후이기 때문에), 후보자스코어(candidatescore)는 무한대값이다. 여기서 상기 후보자스코어는 상기 후보자리스트에 저장되어 있는 엔트리 개수가 k(2)개 미만(E4만 저장(1개 저장))이기 때문에, 다시 무한대값을 유지한다.
- [0125] 상기 서치리스트에서 삭제한 엔트리 E3가 가리키는 노드 N3에 접근하여 읽어들인 후에는, 상기 노드 N3가 단말 노드인지 비단말 노드인지 판단한다. 상기 노드 N3는 단말 노드이다.
- [0126] 따라서, 테이터를 가리키는 상기 단말 노드 N3에 포함되는 각 자식 엔트리에 대하여, 각 자식 엔트리가 가리키

는 데이터의 스코어가 후보자스코어보다 큰지 판단한다. 그런 다음, 데이터의 스코어가 후보자스코어보다 크지 않는 경우에만, 상기 데이터를 가리키는 자식 엔트리만 후보자리스트에 삽입 저장한다.

- [0127] 상기 각 자식 엔트리 중, 첫 번째 엔트리가 가리키는 데이터인 r1의 데이터 스코어는 13.07이고 후보자스코어는 무한대값이므로, 상기 r1의 데이터 스코어는 상기 후보자스코어보다 크지 않다. 따라서, 상기 데이터 r1을 가리키는 자식 엔트리를 후보자리스트에 삽입 저장한다.
- [0128] 이때, 상기 서치리스트(search list)에는 {E4, E2}을 저장하고 있고, 후보자리스트(candidatelist)에는 {r1을 가리키는 자식 엔트리, E4}을 저장하고 있으며, 후보자스코어(candidatescore)는 13.6의 값이다. 여기서 상기 후보자리스트에 저장되어 있는 데이터 r1의 스코어는 13.07이고, E4의 최대스코어는 13.6이기 때문에, 데이터의 스코어 또는 최대스코어의 오름차순으로 정렬될 수 있도록 저장되고, 상기 후보자스코어는 상기 후보자리스트에 저장되어 있는 엔트리 개수가 k(2)개이기 때문에, k번째 엔트리 E4의 최대스코어인 13.6의 값이 된다.
- [0129] 다음, 상기 각 자식 엔트리 중, 두 번째 엔트리가 가리키는 데이터인 r2의 데이터 스코어는 9.07이고 후보자스코어는 13.6의 값이므로, 상기 r2의 데이터 스코어는 상기 후보자스코어보다 크지 않다. 따라서, 상기 데이터 r2을 가리키는 자식 엔트리를 후보자리스트에 삽입 저장한다.
- [0130] 이때, 상기 서치리스트(search list)에는 {E4, E2}을 저장하고 있고, 후보자리스트(candidatelist)에는 {r2를 가리키는 자식 엔트리, r1을 가리키는 자식 엔트리}을 저장하고 있으며, 후보자스코어(candidatescore)는 13.07의 값이다.
- [0131] 여기서 상기 후보자리스트에 저장될 수 있는 엔트리의 개수는 k개(2개)이다. 따라서, r2를 가리키는 자식 엔트리가 삽입 저장되기 전에 k번째에 저장되어 있던 엔트리 E4를 삭제한 후, 엔트리들의 최대스코어 또는 데이터의 스코어가 오름차순으로 정렬될 수 있도록 삽입저장한다. 결과적으로, 상기 후보자리스트에는 {r2를 가리키는 자식 엔트리, r1을 가리키는 자식 엔트리}가 저장되고, 후보자스코어는 데이터 r1의 스코어인 13.07의 값이 된다.
- [0132] 다음, 서치리스트가 비워진 상태가 아니기 때문에, 서치리스트에서 최상위에 위치한 엔트리인 E4를 삭제하고, 상기 서치리스트에서 삭제된 엔트리 E4의 최소스코어가 상기 후보자스코어보다 큰지 판단한다. 상기 서치리스트에서 삭제된 엔트리 E4의 최소스코어는 2(표 2 참조)이고 후보자스코어는 13.07이므로, 서치리스트에서 삭제된 엔트리 E4의 최소스코어가 후보자스코어보다 크지 않다.
- [0133] 따라서, 상기 서치리스트에서 삭제된 엔트리 E4가 가리키는 노드 N4를 접근하여 읽어들인다. 한편, 상기 서치리스트에서 삭제된 엔트리 E4는 상기 후보자리스트에 존재하지 않기 때문에, 상기 후보자리스트에서 E4를 삭제하는 과정은 생략된다.
- [0134] 이때, 상기 서치리스트(search list)에는 {E2}을 저장하고 있고(E4가 삭제된 후이기 때문에), 후보자리스트(candidatelist)에는 {r2를 가리키는 자식 엔트리, r1을 가리키는 자식 엔트리}을 저장하고 있으며, 후보자스코어(candidatescore)는 13.07의 값이다.
- [0135] 상기 서치리스트에서 삭제한 엔트리 E4가 가리키는 노드 N4에 접근하여 읽어들인 후에는, 상기 노드 N4가 단말 노드인지 비단말 노드인지 판단한다. 상기 노드 N4는 단말 노드이다.
- [0136] 따라서, 데이터를 가리키는 상기 단말 노드 N4에 포함되는 각 자신 엔트리에 대하여, 각 자식 엔트리가 가리키는 데이터의 스코어가 후보자스코어보다 큰지 판단한다. 그런 다음, 데이터의 스코어가 후보자스코어보다 크지 않는 경우에만, 상기 데이터를 가리키는 자식 엔트리만 후보자리스트에 삽입 저장한다.
- [0137] 상기 각 자식 엔트리 중, 첫 번째 엔트리가 가리키는 데이터인 r3의 데이터 스코어는 12(표 1 참조)이고 후보자스코어는 13.07이므로, 상기 r3의 데이터 스코어는 상기 후보자스코어보다 크지 않다. 따라서, 상기 데이터 r3을 가리키는 자식 엔트리를 후보자리스트에 삽입 저장한다.
- [0138] 이때, 상기 서치리스트(search list)에는 {E2}을 저장하고 있고, 후보자리스트(candidatelist)에는 {r2를 가리키는 자식 엔트리, r3을 가리키는 자식 엔트리}을 저장하고 있으며, 후보자스코어(candidatescore)는 12의 값이다.
- [0139] 여기서 상기 후보자리스트에 저장될 수 있는 엔트리의 개수는 k개(2개)이다. 따라서, r3를 가리키는 자식 엔트리가 삽입 저장되기 전에 k번째에 저장되어 있던 r1을 가리키는 자식 엔트리를 삭제한 후, 엔트리들의 최대스코어 또는 데이터의 스코어가 오름차순으로 정렬될 수 있도록 삽입저장한다. 결과적으로, 상기 후보자리스트에는 {r2를 가리키는 자식 엔트리, r3을 가리키는 자식 엔트리}가 저장되고, 후보자스코어는 데이터 r3의 스코어인

12의 값이 된다.

- [0140] 다음, 상기 각 자식 엔트리 중, 두 번째 엔트리가 가리키는 데이터인 r4의 데이터 스코어는 14.84(표1 참조)이고 후보자스코어는 12의 값이므로, 상기 r4의 데이터 스코어는 상기 후보자스코어보다 크다. 따라서, 더 이상 상기 r4를 고려하지 않고 다음 절차를 진행한다.
- [0141] 다음, 서치리스트가 비워진 상태가 아니기 때문에, 현재 서치리스트에 저장되어 있는 엔트리들 중에서, 최상위에 위치한 엔트리 E2를 삭제하고, 상기 서치리스트에서 삭제된 엔트리 E2의 최소스코어가 상기 현재 후보자스코어보다 큰지 판단한다.
- [0142] 상기 서치리스트에서 삭제된 엔트리 E2의 최소스코어는 0(표 2 참조)이고, 상기 후보자스코어는 12이므로, 상기 서치리스트에서 삭제된 엔트리 E2의 최소스코어는 후보자스코어보다 크지 않다. 따라서, 상기 서치리스트에서 삭제된 엔트리 E2가 가리키는 노드 N2를 접근하여 읽어들인다. 이 때, 상기 서치리스트에서 삭제된 엔트리 E2는 상기 후보자리스트에 존재하지 않기 때문에, 서치리스트에서 삭제된 엔트리를 후보자리스트에서 삭제하는 과정은 생략된다.
- [0143] 이 때, 상기 서치리스트(search list)에는 { }을 저장하고 있고(E2가 삭제되었기 때문에), 후보자리스트(candidate list)에는 {r2을 가리키는 자식 엔트리, r3을 가리키는 자식 엔트리}을 저장하고 있으며, 후보자스코어(candidatescore)는 12의 값이다.
- [0144] 다음, 상기 서치리스트에서 삭제된 엔트리 E2가 가리키는 노드 N2가 단말 노드인지, 비단말 노드인지 판단한다. 상기 노드 N2은 비단말 노드이다. 따라서, 상기 비단말 노드 N2에 포함되는 각 자식 엔트리(E5, E6)에 대하여, 각 자식 엔트리의 최소스코어가 후보자스코어보다 큰지 판단한다.
- [0145] 먼저, 상기 비단말 노드 N2에 포함되는 자식 엔트리 E5에 대하여 판단하면, 상기 자식 엔트리 E5의 최소스코어는 5.65이고(표 2 참조), 후보자스코어는 12값이다. 따라서, 상기 자식 엔트리 E5의 최소스코어가 후보자스코어보다 크지 않고, 결과적으로 상기 자식 엔트리 E5를 서치리스트에 삽입한다.
- [0146] 하지만, 상기 E5는 후보자리스트에 삽입 저장하지는 않는다. 후보자리스트에 저장되어 있는 엔트리의 개수가 k개인 경우에는 상기 자식 엔트리의 최대스코어가 상기 후보자스코어보다 작은 경우에만 상기 후보자리스트의 k번째 엔트리를 삭제한 후, 엔트리들의 최대스코어가 오름차순으로 정렬될 수 있도록 삽입저장되기 때문이다.
- [0147] 상기, 자식 엔트리 E5의 최대스코어는 15.07(표 3 참조)이고 후보자스코어는 12이므로, 상기 자식 엔트리 E5의 최대스코어는 상기 후보자스코어보다 크다. 따라서, 상기 자식 엔트리인 E5는 후보자리스트에 저장되지 않는다.
- [0148] 이 때, 상기 서치리스트(search list)에는 {E5}을 저장하고 있고, 후보자리스트(candidate list)에는 {r2을 가리키는 자식 엔트리, r3을 가리키는 자식 엔트리}을 저장하고 있으며, 후보자스코어(candidatescore)는 12의 값이다.
- [0149] 마찬가지로, 상기 비단말 노드 N2에 포함되는 자식 엔트리 E6에 대하여 판단하면, 상기 자식 엔트리 E6의 최소스코어는 3이고(표 2 참조), 후보자스코어는 12값이다. 따라서, 상기 자식 엔트리 E6의 최소스코어가 후보자스코어보다 크지 않고, 결과적으로 상기 자식 엔트리 E6를 서치리스트 및 후보자리스트에 삽입한다.
- [0150] 이 때, 상기 자식 엔트리 E6의 최대스코어는 5.6이고, 상기 후보자스코어는 12이기 때문에, 상기 자식 엔트리 E6의 최대스코어는 상기 후보자스코어보다 크지 않다. 따라서, 상기 자식 엔트리 E6을 후보자리스트에 삽입 저장하는 것이다.
- [0151] 지금까지, 상기 서치리스트(search list)에는 {E5, E6}을 저장하고 있고, 후보자리스트(candidate list)에는 {E6, r2을 가리키는 자식 엔트리}을 저장하고 있으며, 후보자스코어(candidatescore)는 9.07의 값이다.
- [0152] 여기서, 상기 서치리스트에 저장되는 엔트리들은 엔트리의 전송 순서대로 정렬하여 삽입 저장된다. 즉, 스트림을 통하여 읽혀지는 순서대로 삽입 저장된다. 그리고, 상기 후보자리스트에는 저장되는 엔트리의 최대스코어가 오름차순으로 정렬될 수 있도록 삽입 저장된다. 즉, 상기 엔트리 E6의 최대스코어는 5.6이고, r2 데이터의 스코어가 9.07이기 때문에, 최대스코어의 오름차순으로 정렬되어 저장될 수 있도록, 상기 후보자리스트에는 {E6, r2을 가리키는 자식 엔트리}가 저장되어 있다. 또한, 상기 후보자스코어는 상기 후보자리스트의 k(클라이언트가 2로 설정함)번째 최대스코어인 9.07의 값이 된다.
- [0153] 다음, 상기 서치리스트가 비워진 상태가 아니기 때문에, 다시 상기 서치리스트에 저장되어 있는 최상위에 위치한 엔트리 E5를 삭제하고, 상기 서치리스트에서 삭제한 엔트리 E5의 최소스코어가 상기 현재 후보자스코어보다

큰지 판단한다.

- [0154] 상기 서치리스트에서 삭제한 엔트리 E5의 최소스코어는 5.65이고, 상기 후보자스코어는 9.07이므로, 상기 서치리스트에서 삭제한 엔트리 E5의 최소스코어가 후보자스코어보다 크지 않다. 따라서, 상기 서치리스트에서 삭제한 엔트리 E5가 가리키는 노드 N5에 접근하여 읽어들인다. 한편, 현재, 후보자리스트에는 상기 E5가 존재하지 않기 때문에, 후보자리스트에서 상기 E5를 삭제하는 과정은 생략한다.
- [0155] 이 때, 상기 서치리스트(search list)에는 {E6}을 저장하고 있고, 후보자리스트(candidatelist)에는 {E6, r2}을 가리키는 자식 엔트리}을 저장하고 있으며, 후보자스코어(candidatescore)는 9.07 값이다.
- [0156] 상기 서치리스트에서 삭제한 엔트리 E5가 가리키는 노드 N5에 접근하여 읽어들인 후에는, 상기 노드 N5가 단말 노드인지 비단말 노드인지 판단한다. 상기 노드 N5는 단말 노드이다.
- [0157] 따라서, 데이터를 가리키는 상기 단말 노드 N5에 포함되는 각 자신 엔트리에 대하여, 각 자식 엔트리가 가리키는 데이터의 스코어가 후보자스코어보다 큰지 판단한다. 그런 다음, 데이터의 스코어가 후보자스코어보다 크지 않는 경우에만, 상기 데이터를 가리키는 자식 엔트리만 후보자리스트에 삽입 저장한다.
- [0158] 상기 각 자식 엔트리 중, 첫 번째 엔트리가 가리키는 데이터인 r5의 데이터 스코어는 15.07이고 후보자스코어는 9.07값이므로, 상기 r5의 데이터 스코어는 상기 후보자스코어보다 크다. 따라서, 상기 r5는 더 이상 고려하지 않고 다음 과정을 수행한다.
- [0159] 다음, 상기 각 자식 엔트리 중, 두 번째 엔트리가 가리키는 데이터인 r6의 데이터 스코어는 13.21이고 후보자스코어는 9.07의 값이므로, 상기 r6의 데이터 스코어는 상기 후보자스코어보다 크다. 따라서, 상기 데이터 r6은 더 이상 고려하지 않고 다음 과정을 수행한다.
- [0160] 지금까지, 상기 서치리스트(search list)에는 {E6}을 저장하고 있고, 후보자리스트(candidatelist)에는 {E6, r2}을 가리키는 자식 엔트리}을 저장하고 있으며, 후보자스코어(candidatescore)는 9.07 값이다.
- [0161] 다음, 서치리스트가 비워진 상태가 아니기 때문에, 서치리스트에서 최상위에 위치한 엔트리인 E6를 삭제하고, 상기 서치리스트에서 삭제된 엔트리 E6의 최소스코어가 상기 후보자스코어보다 큰지 판단한다. 상기 서치리스트에서 삭제된 엔트리 E6의 최소스코어는 3(표 2 참조)이고 후보자스코어는 9.07이므로, 서치리스트에서 삭제된 엔트리 E6의 최소스코어가 후보자스코어보다 크지 않다.
- [0162] 따라서, 상기 서치리스트에서 삭제된 엔트리 E6가 가리키는 노드 N6를 접근하여 읽어들인다. 한편, 상기 서치리스트에서 삭제된 엔트리 E6는 상기 후보자리스트에도 존재하기 때문에, 상기 후보자리스트에서 E6를 삭제한다.
- [0163] 이때, 상기 서치리스트(search list)에는 { }을 저장하고 있고, 후보자리스트(candidatelist)에는 {r2}을 가리키는 자식 엔트리}을 저장하고 있으며, 후보자스코어(candidatescore)는 무한대값이다. 상기 후보자스코어는 상기 후보자리스트에 저장된 엔트리 개수가 k개(2개) 미만이기 때문에 다시 무한대값으로 유지한다.
- [0164] 상기 서치리스트에서 삭제한 엔트리 E6가 가리키는 노드 N6에 접근하여 읽어들인 후에는, 상기 노드 N6가 단말 노드인지 비단말 노드인지 판단한다. 상기 노드 N6는 단말 노드이다.
- [0165] 따라서, 데이터를 가리키는 상기 단말 노드 N6에 포함되는 각 자신 엔트리에 대하여, 각 자식 엔트리가 가리키는 데이터의 스코어가 후보자스코어보다 큰지 판단한다. 그런 다음, 데이터의 스코어가 후보자스코어보다 크지 않는 경우에만, 상기 데이터를 가리키는 자식 엔트리만 후보자리스트에 삽입 저장한다.
- [0166] 상기 각 자식 엔트리 중, 첫 번째 엔트리가 가리키는 데이터인 r7의 데이터 스코어는 3.6(표 1 참조)이고 후보자스코어는 9.07이므로, 상기 r7의 데이터 스코어는 상기 후보자스코어보다 크지 않다. 따라서, 상기 데이터 r7을 가리키는 자식 엔트리를 후보자리스트에 삽입 저장한다.
- [0167] 이때, 상기 서치리스트(search list)에는 { }을 저장하고 있고, 후보자리스트(candidatelist)에는 {r7}을 가리키는 자식 엔트리, r2를 가리키는 자식 엔트리}을 저장하고 있으며, 후보자스코어(candidatescore)는 9.07의 값이다.
- [0168] 다음, 상기 각 자식 엔트리 중, 두 번째 엔트리가 가리키는 데이터인 r8의 데이터 스코어는 7.38(표1 참조)이고 후보자스코어는 9.07의 값이므로, 상기 r8의 데이터 스코어는 상기 후보자스코어보다 크지 않다. 따라서, 상기 데이터 r8을 가리키는 자식 엔트리를 후보자리스트에 삽입 저장한다.
- [0169] 이때, 상기 서치리스트(search list)에는 { }을 저장하고 있고, 후보자리스트(candidatelist)에는 {r7}을 가리키는

는 자식 엔트리, r8을 가리키는 자식 엔트리)을 저장하고 있으며, 후보자스코어(candidatescore)는 7.38의 값이다. 여기서, 상기 r8 데이터의 스코어는 7.38이기 때문에, 후보자스코어가 7.38이 된다.

[0170] 이상과 같이, 반복적인 절차를 수행한 결과, 서치리스트가 비워진 상태가 된다. 따라서, 후보자리스트에 삽입되어 있는 엔트리가 가리키는 데이터들을 수신하여 결과를 도출한다. 즉, r7 및 r8이 가리키는 단말 엔트리를 통해 데이터 r7 및 r8을 읽어들인다. 최종 결과는 r7 및 r8이 된다.

[0171] 이상에서 설명한 방법에 의하여 무선 데이터 방송 스트림에서 위치 기반 상위-k 질의 처리를 효과적으로 수행할 수 있다. 다만, 이상에서 설명한 처리 위치 기반 상위-k 질의 처리 방법에는 한가지 단점을 가지고 있다.

[0172] 이상에서 살펴본 바와 같이, 위치기반 상위-k 질의 q가 주어졌을 때, 방송 스트림 상의 aR-tree 탐색 알고리즘은 최소스코어(lowerscore)가 후보자스코어(candidate score)보다 크기만 하면, 모든 aR-tree의 비단말 엔트리만을 고려 대상에서 제외시킨다.

[0173] 하지만, 각 비단말 엔트리 E는 대부분 큰 범위의 1차원 구간 $E.ivl = [\underline{ivl}, \overline{ivl}]$ 을 포함한다. 따라서 $\underline{ivl} \leq d.A \leq \overline{ivl}$ 인 경우가 많이 발생하며, mindist(q.V, E.ivl)의 값은 0이 된다. 이는 부정확하게 작은 최소스코어(lowerscore(q, E)) 값을 초래하게 되며, 필요 없는 노드의 접근 수를 증가시킨다.

[0174] 더욱이, 각 비단말 엔트리 E는 $\underline{ivl}$ 값과 $\overline{ivl}$ 값을 표현하기 위해 2개의 부동 소수점 값을 포함해야 하고, 이는 E의 사이즈 증가를 초래한다. Fmax = (Bucket Size)/(Size of E)를 aR-tree 노드의 최대 팬-아웃(fan-out)이라고 하자. E의 사이즈가 증가하게 되면, Fmax 또한 증가하게 된다.

[0175] 이는 aR-tree 높이의 증가를 초래하게 되며 알고리즘의 평균 노드 접근 수를 증가시킨다. 즉 알고리즘의 튜닝 시간 성능을 저하시킨다. 또한 방송 스트림의 길이가 증가하게 되므로 알고리즘의 평균 접근 시간 성능도 저하시킨다.

[0176] 위와 같은 단점을 극복하기 위하여 본 발명에서는 또 다른 R-tree의 변형인 비트벡터(bit-vector) R-tree(bR-tree)를 제2 실시예로 설명한다.

[0177] bR-tree의 각 비단말 엔트리 E는 subtree(E)에 포함된 모든 데이터의 비공간 속성 값들을 적은 수의 비트로 포괄적으로 표현하기 위한 비트벡터를 포함한다. 특히, subtree(E)에 포함된 데이터의 비공간 속성 값들을 비트벡터로 부호화하기 위해 비공간 속성 A의 영역을 다수의 독립된 클러스터(cluster)로 나눈다.

[0178] $N(\leq |D|)$ 을 D가 포함하는 서로 다른 비공간 속성 값의 총 개수라고 하자. 이때, 서로 다른 비공간 속성 값을 K-평균 알고리즘(K-means algorithm)을 활용하여 K개의 클러스터로 나눌 수 있다. 그 후, 각 비단말 엔트리 E는 다음과 같은 노드 비트벡터(node bit-vector)를 포함한다.

[0179] 각 클러스터 $c_i(1 \leq i \leq K)$ 에 대해 하나의 비트를 할당한다고 하자. 이 때, 각 비단말 엔트리 E가 포함하는 노드 비트벡터 E.bv는 K개의 비트로 이루어진 비트열이며, 각 클러스터에 해당하는 비트 $b_i(1 \leq i \leq K)$ 는 $(\exists d \in subtree(E): d.A \in c_i)$ 조건을 만족할 경우에만 '1' 값을 가진다.

[0180] 한편, bR-tree의 각 단말 엔트리는 aR-tree와 마찬가지로 자신이 가리키는 데이터의 비공간 속성 값을 추가적으로 저장한다. 도 7은 도 2에서 보여준 레스토랑 데이터를 색인하는 bR-tree를 보여준다(K=4).

[0181] 예를 들어, E5.mbr은 레스토랑 r5, r6을 공간적으로 포함하며, 이들의 평균 메뉴 가격은 각각 \$18, \$4이므로 클러스터 c4, c1에 포함된다. 따라서 E5의 노드 비트벡터 E5.bv = '1001'이다.

[0182] 도 6을 통해서 각각의 클러스터 $c_i(1 \leq i \leq 4)$ 는 자신이 포함한 평균메뉴 가격들의 최소, 최대값으로 이루어진 독립된 1차원 클러스터 구간을 나타낼 수 있음을 관찰할 수 있으며(예를 들어, $c_1 \rightarrow [4, 5]$, $c_4 \rightarrow [18, 20]$), E5.bv를 통해 모든 레스토랑 $r \in subtree(E_5)$ 의 평균메뉴 가격은 클러스터 구간 [4, 5] 혹은 [18, 20]에 속함을 알 수 있다.

[0183] bR-tree의 각 비단말 엔트리 E가 포함하는 노드 비트벡터 E.bv가 주어졌을 때, 자신에 해당하는 비트 b가 '1' 값을 가진 클러스터 c의 집합을 $C(|C| \leq K)$ 라고 하자. 이 때, 각각의 $c \in C$ 는 자신이 포함하는 비공간 속성 값들

의 최소값 $\underline{c}$, 최대값 $\bar{c}$ 으로 이루어진 클러스터 구간 $[\underline{c}, \bar{c}]$ 을 나타낼 수 있으며, E.bv를 통해 모든 데이터 $d \in \text{subtree}(E)$ 의 비공간 속성 값 d.A는 각각의 $c \in C$ 가 나타낼 수 있는 클러스터 구간들 중 하나에 속함을 알 수 있다.

[0184] 이와 같은 bR-tree를 정리하면, 비단말 노드의 자식 엔트리는 앞서 설명한 실시예 1에서 규정한 aR-tree의 구간 값 대신에 비트벡터를 추가적으로 저장하고 있다.

[0185] 상기 비트벡터는 K개의 비트로 이루어진 비트열이고, 각 비트는 K개의 클러스터에 각각 일대일 대응되고, 상기 클러스터 각각은 상기 비공간 속성 값들을 독립적인 구간으로 나누기 위한 클러스터 최소값과 클러스터 최대값으로 이루어진 클러스터 구간값을 가진다. 그리고, 상기 비단말 노드의 자식 엔트리가 저장하고 있는 비트벡터는 상기 비단말 노드의 자식 엔트리에 포함되는 데이터들의 비공간 속성값들을 포함하는 클러스터 구간값을 가지는 클러스터에 대응되는 비트만 '1'의 값을 가진다.

[0186] 도 7을 참조하여 예를 들면, 비단말 노드의 자식 엔트리 각각은 K가 4인 비트로 이루어진(4비트로 이루어진) 비트열로 구성된 비트벡터를 저장하고 있다. 예를 들어, 자식 엔트리 E2는 '1101'의 비트벡터를 저장하고 있다.

[0187] 그리고, 비공간 속성 값들을 K(=4)개의 클러스터로 나누었다. 여기서, 클러스터 c_1 의 클러스터 구간 값은 [4, 5], 클러스터 c_2 의 클러스터 구간 값은 [8, 10], 클러스터 c_3 의 클러스터 구간 값은 [12, 16], 클러스터 c_4 의 클러스터 구간 값은 [18, 20]이다.

[0188] 상기 각 비트벡터의 각각의 비트는 상기 K개의 클러스터에 각각 일대일 대응된다. 즉, 클러스터 c_1 은 각 비트벡터의 첫번째비트(최상위 비트)에 대응되고, 클러스터 c_2 은 각 비트벡터의 첫번째 비트클러스터 c_3 은 각 비트벡터의 첫번째 비트에 대응되고에 대응되고, 클러스터 c_4 은 각 비트벡터의 네번째 비트(최하위 비트)에 대응된다.

[0189] 도 7에서 자식 엔트리 E5는 비트벡터 '1001'을 저장하고 있다. 이는 자식 엔트리 E5가 포함하고 있는 데이터 r5의 비속성 공간 값인 \$18이 클러스터 c_4 의 클러스터 구간값에 포함되고, 데이터 r6의 비속성 공간 값인 \$4가 클러스터 c_1 의 클러스터 구간값에 포함됨을 알 수 있다.

[0190] 상기와 같은 bR-tree는 방송 스트림 상에 깊이 우선 방식으로 배치된다. 이와 같은 방송 스트림 상에 깊이우선 방식으로 배치되어 있는 bR-tree에서의 탐색 방법 역시 순차적 탐색 방식을 따른다. 또한, 제1 실시예에서 설명한 질의 처리 방법을 그대로 따른다.

[0191] 다만, bR-tree를 방송 스트림 상에 배치하여 방송되는 스트림에서 위치 기반 상위-k 질의 처리를 더 효율적으로 처리하기 위하여, 즉 탐색 범위를 줄이기 위하여 제1 실시예에서 정의한 최소스코어(lower score) 및 최대스코어(upper score)가 변경되어 이용된다. 그 변경된 최소스코어 및 최대스코어에 대하여 이하에서 정의한다.

[0192] 위치기반 상위-k 질의 q와 bR-tree의 비단말 엔트리 E가 주어졌을 때,
$$\underline{c}_{\min} = \min_{c \in C} [q.V - \underline{c}], \overline{c}_{\min} = \min_{c \in C} [q.V - \bar{c}]$$
 라고 하자. 이때, q에 대한 E의 최소스코어(lower score)는 다음과 같은 (수식 4)로 정의된다.

[0193] $\text{lowerscore}(q, E) = \text{mindist}(q.V, E.mbr) + \text{mindist}(q.V, E.bv)$ (수식 4)

[0194] 여기서,

[0195]
$$\text{mindist}(q.V, E.bv) = \begin{cases} \min(\underline{c}_{\min}, \overline{c}_{\min}), & \text{if } \nexists c \in C: \underline{c} \leq q.V \leq \bar{c}; \\ 0 & \text{otherwise} \end{cases}$$

[0196] 위치기반 상위-k 질의 q와 bR-tree의 비단말 엔트리 E가 포함하는 노드 비트벡터 E.bv가 주어졌을 때, $\forall d \in \text{subtree}(E)$, $\text{mindist}(q.V, E.bv) \leq \text{dist}(q.V, d.A)$ 임을 보장할 수 있다.

[0197] 위치기반 상위-k 질의 q와 bR-tree의 비단말 엔트리 E가 주어졌을 때,
$$\underline{c}_{\max} = \max_{c \in C} [q.V - \underline{c}], \overline{c}_{\max} = \max_{c \in C} [q.V - \bar{c}]$$
 라고 하자. 이때, q에 대한 E의 최대스코어(upper score)는 다음과 같이 (수식 5)로 정의된다.

[0198] $\text{upperscore}(q, E) = (\text{minmaxdist}(q, P, E, \text{mbr}) + \text{maxdist}(q, V, E, \text{bv}))$ (수식 5)

[0199] 여기서, $\text{maxdist}(q, V, E, \text{bv}) = \max(\underline{c}_{\text{max}}, \overline{c}_{\text{max}})$ 이다.

[0200] 위치기반 상위-k 질의 q와 bR-tree의 비단말 엔트리 E가 포함하는 노드 비트벡터 E.bv가 주어졌을 때, $\forall d \in \text{subtree}(E), \text{mindist}(q, V, E, \text{bv}) \leq \text{dist}(q, V, d, A)$ 임을 보장할 수 있다.

[0201] 이와 같이 정의되는 최소스코어 및 최대스코어를 이용하여 무선 데이터 방송 스트림에서 위치 기반 상위-k 질의 처리를 수행할 수 있다. 그 처리 방법은 앞에서 설명한 실시예 1과 대부분 비슷하므로 자세한 설명은 생략한다. 즉, 최소스코어 및 최대스코어를 계산하는 수식과, 각 자식 엔트리는 비트벡터를 저장하고 있다는 점만 다를 뿐, 탐색 방법은 aR-tree와 동일하다.

[0202] 두 개의 자료 구조인 서치리스트(search list) 및 후보자리스트(candidate list)를 사용하여 위치 기반 상위-k 질의 처리 방법은 후보자스코어(candidate score)보다 작은 최소스코어를 가지는 엔트리가 가리키는 노드만을 순차적으로 접근한다.

[0203] 여기서 주목할 점은 노드 비트벡터는 다수의 작은 범위 구간(클러스터 구간)을 나타낼 수 있으므로, aR-tree에 비해 더 정확한 최소스코어를 제공할 수 있다. 따라서 더 나은 탐색 성능을 제공한다.

[0204] 이하에서는 본 발명의 제1 실시예인 aR-tree 기반 기법과 제2 실시예인 bR-tree 기반 기법의 성능을 튜닝 시간과 접근 시간 측면에서 살펴본다.

[0205] 성능 평가를 위한 하드웨어 플랫폼으로는 2GB의 메모리를 장착한 팬티엄 IV, 3.19GHz의 PC를 사용하였다. 또한 소프트웨어 플랫폼으로는 Windows XP 운영체제 및 자바(JAVA) 언어를 사용하였다. 시스템 모델은 하나의 서버, 하나의 방송 채널 및 다수의 이동 사용자로 구성되어 있다[B. Zheng, W.-C. Lee and D. Lee, "Spatial Queries in Wireless Broadcast Systems," Wireless Network, 10 (6), 2004. 참조].

[0206] 성능 평가를 위해 5 개의 데이터셋 (dataset)을 생성하였으며, 2차원 단위 공간 $[0, 1]^2$ 에 균일한 분포(uniform distribution)로 위치시켰다. 또한 데이터셋의 각 데이터 d는 3 개의 비공간 속성 값 $d.A_1, d.A_2, d.A_3$ 을 포함하며, 이들은 쏘림 계수 $\Theta=0.8$ 를 가지는 지프 분포(zipf distribution)를 이용하여 $[0, 1]$ 사이의 실수 값을 가지도록 생성하였다. 위치기반 상위-k 질의도 비슷하게 생성하였다.

[0207] 특히, 각 질의 q는 사용자의 위치와 데이터의 비공간 속성에 대한 조건으로 지정한 3개의 값을 명시하도록 하였

$$\text{dist}(q, V, d, A) = \sum_{i=1}^3 |q.V_i - d.A_i|$$

다. 질의 q와 d의 공간 거리 $\text{dist}(q, V, d, A)$ 와 비공간 거리 는 같은 범위의 값을 가지도록 정규화 시켰다.

[0208] 정확한 성능 평가를 위하여 실험별로 1,000번의 질의를 생성하였으며 평균 튜닝 시간과 접근 시간을 버킷의 수로 측정하였다. 표 1은 실험에서 사용한 실험 변수 및 값을 보여주고 있으며, 볼드체는 기본 값을 나타낸다.

Parameter	setting
데이터셋	10K, 20K, 30K, 40K, 50K
k	1, 8, 16, 32, 64, 128, 256
버킷 용량	128, 256, 512, 1024, 2048(bytes)

[0210] - 표 1 -

[0211] Sort Tile Recursive(STR) packing 기법[S.T. Leutenegger, J.M. Edgington and M.A. Lopez, "STR: A Simple and Efficient Algorithm for R-tree Packing," In ICDE, 1997.]을 사용하여 aR-tree와 bR-tree를 구축하였으며, 비공간 속성 정보를 위한 연산을 추가하였다.

[0212] 또한, (1,m) 색인 인터리빙 기법[T.Imielinski, S.Viswanathan and B.R.Bardrinath, "Data on Air: Organization and Access," IEEE Trans. Knowledge and Data Engineering, 9(3), 1997.]을 사용하여 방송 스트림에 m번 중복/배치하였다.

[0213] 각 데이터는 1개의 데이터 버킷에 해당한다고 가정하였다. aR-tree의 비단말 엔트리 E는 2 bytes의 포인터, 16

bytes의 공간 속성 정보(E.mbr)와 24 bytes(각 비공간 속성 당 8 bytes)의 비공간 속성 정보(E.ivl)를 포함한다.

- [0214] 한편, bR-tree는 비단말 엔트리 E에 포함된 노드 비트벡터 E.bv의 사이즈(즉, K의 값)에 따라 접근 시간 및 튜닝 시간 성능에 큰 영향을 미친다. 따라서 각 비공간 속성 $A_i (1 \leq i \leq 3)$ 에 할당된 노드 비트벡터 E.bv_i의 사이즈에 따른 성능을 평가해 보았다.
- [0215] 도 8의 왼쪽 그래프에서 보이듯이, 노드 비트벡터 E.bv_i의 사이즈가 증가할수록 접근시간은 점점 증가한다. 그 이유는 bR-tree의 사이즈가 증가하게 되므로 방송 스트림의 길이 또한 증가하기 때문이다.
- [0216] 튜닝 시간 측면의 경우, 도 8의 오른쪽 그래프에서 보이듯이, E.bv_i의 사이즈가 3 bytes일 때 가장 좋은 성능을 보인다. 그 이유는 큰 사이즈의 E.bv_i의 경우는 bR-tree의 높이가 증가하므로 노드의 접근 수를 증가시키기 때문이다.
- [0217] 한편, 작은 사이즈의 E.bv_i의 경우 bR-tree의 높이를 줄일 수 있지만, 비공간 속성 정보가 부족하여 필요없는 노드의 접근 수를 증가시킨다. 노드 비트벡터 접근 수를 증가시킨다. 노드 비트벡터 E.bv_i의 사이즈가 3bytes일 때 튜닝 시간 성능뿐만 아니라 접근 시간 성능 측면에서도 좋은 결과를 보이므로, 이후의 실험부터 3 bytes의 E.bv_i를 사용 하였다.
- [0218] 첫 번째 실험은 데이터셋의 사이즈(데이터의 수)에 따른 aR-tree 기반 기법과 bR-tree 기반 기법의 성능을 비교해 본다.
- [0219] 도 9에서 보이듯이, 두 기법 모두 데이터의 수가 증가할수록 접근 시간과 튜닝 시간이 증가하는 것을 확인할 수 있다. 그 이유는 데이터의 수가 증가할수록 aR-tree와 bR-tree의 사이즈도 증가하게 되며, 탐색 알고리즘은 더 많은 수의 aR-tree/bR-tree 노드에 접근해야하기 때문이다.
- [0220] 하지만, bR-tree 기반 기법이 aR-tree기반 기법에 비해 더 좋은 접근 및 튜닝 시간 성능을 보인다. 그 이유는 aR-tree에 비해 bR-tree의 사이즈가 더 작으며, 더 나은 탐색 성능을 제공하기 때문이다. aR-tree 기반 기법과 비교하여 bR-tree 기반 기법은 평균 97.6%의 접근 시간, 82.2%의 튜닝 시간을 소모한다.
- [0221] 도 10은 k의 변화에 따른 두 기법의 접근 시간과 튜닝시간 성능을 보이고 있다. 첫 번째 실험과 마찬가지로 두 번째 실험에서도 bR-tree 기반 기법이 aR-tree기반 기법에 비해 더 좋은 성능을 보이고 있다. aR-tree 기반 기법과 비교하여 bR-tree 기반 기법은 평균 97.4%의 접근 시간, 77.4%의 튜닝 시간을 소모한다.
- [0222] 마지막으로 버킷 용량의 변화에 따른 두 기법의 접근 시간과 튜닝 시간 성능을 비교해 본다. 도 11에서 보이듯이, 두 기법 모두 버킷의 용량이 늘어날수록 접근 시간과 튜닝시간이 감소하는 것을 확인할 수 있다. 그 이유는 버킷 용량이 늘어날수록 aR-tree/bR-tree의 높이도 감소하고, 방송 스트림의 길이도 더 줄어들기 때문이다.
- [0223] 본 발명은 무선 데이터 방송 시스템에서 사용자가 명시한 공간/비공간 조건과 가장 잘 부합되는 상위 k 개의 데이터를 찾는 새로운 위치기반 질의인 위치기반 상위-k 질의의 수행에 대해 설명하였다.
- [0224] 무선 방송 스트림 상에서 위치기반 상위-k 질의를 수행하기 위한 첫 번째 기법으로서 우리는 aR-tree 기반의 기법을 소개하였다. 그 후, 각 비단말 엔트리에 비트 벡터 정보를 포함하는 새로운 R-tree의 변형인 bR-tree를 제안하였다.

도면의 간단한 설명

- [0225] 도 1은 기존의 (1, m) 색인 인터리빙 기법의 예시도이다.
- [0226] 도 2는 본 발명에서 사용할 샘플 지역 레스토랑 데이터를 보여주는 예시도이다.
- [0227] 도 3은 R-tree 색인 구조를 보여주는 예시도이다.
- [0228] 도 4는 분기한정 탐색 방식과 방송 스트림의 순차적 특성을 보여주는 예시도이다.
- [0229] 도 5는 본 발명에 적용되는 aR-tree 색인 구조를 보여주는 예시도이다.
- [0230] 도 6은 본 발명을 수행하기 위한 절차도이다.

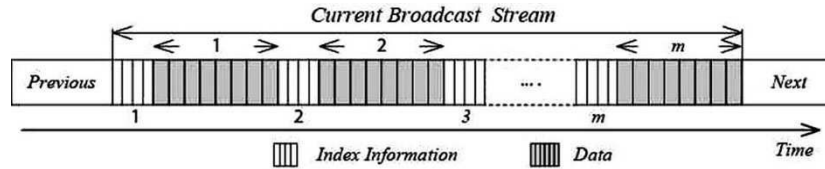
[0231] 도 7은 본 발명에 적용되는 bR-tree 색인 구조를 보여주는 예시도이다.

[0232] 도 8 및 도 9는 본 발명의 성능 평가 결과인 노드 비트벡터 사이즈 대 접근 시간 및 튜닝시간 그래프이다.

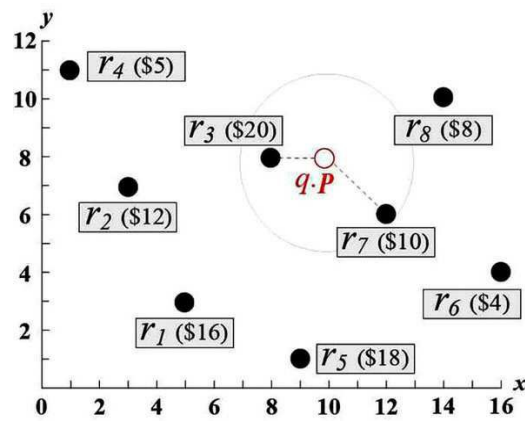
[0233] 도 10 및 도 11은 본 발명의 성능 평가 결과인 K 대 접근 시간 및 튜닝시간 그래프이다.

도면

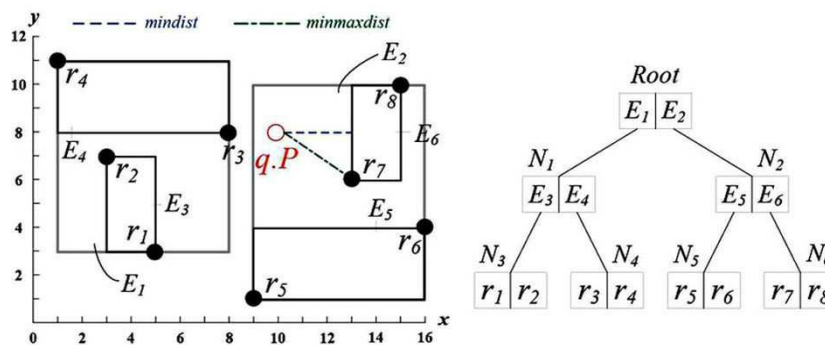
도면1



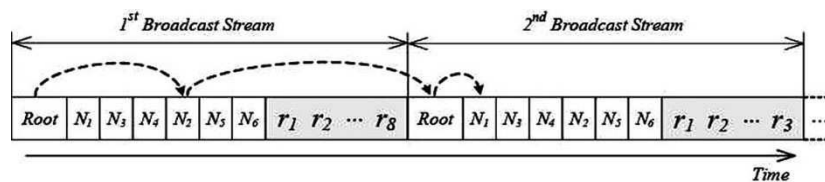
도면2



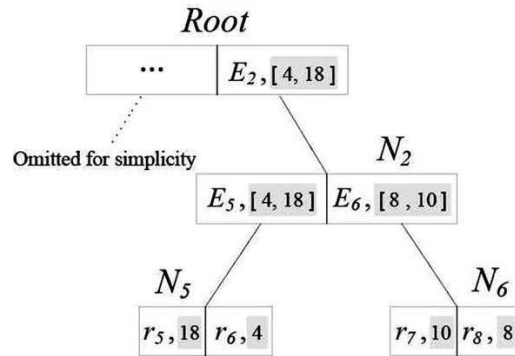
도면3



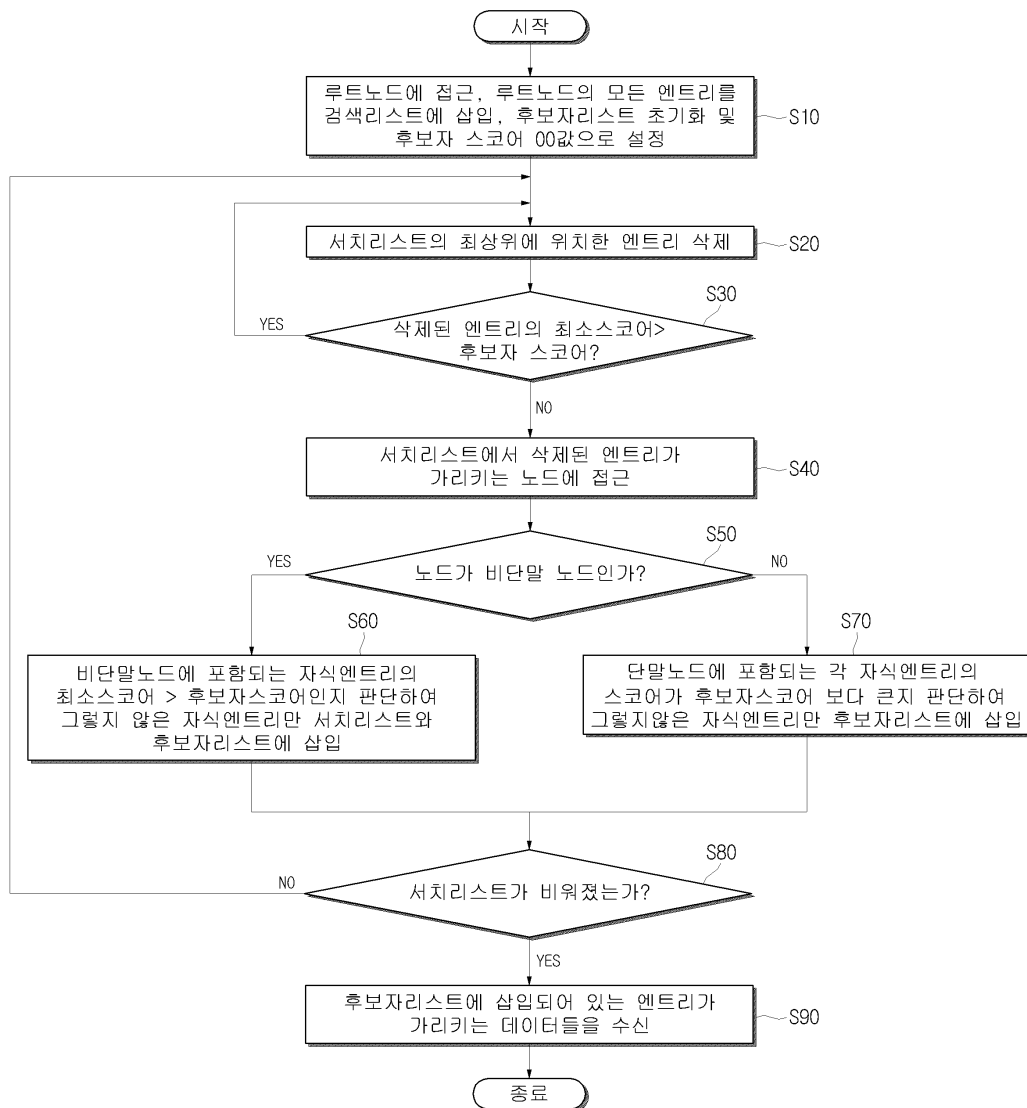
도면4



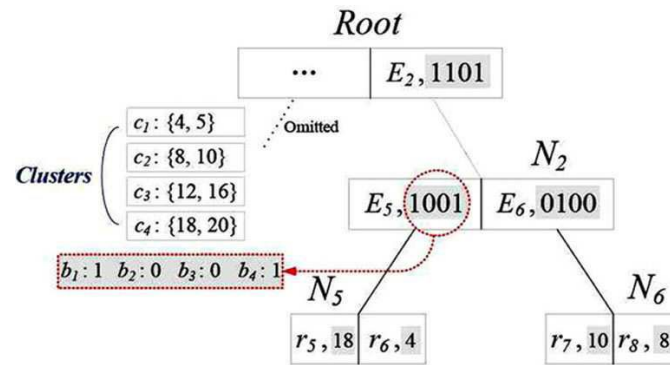
도면5



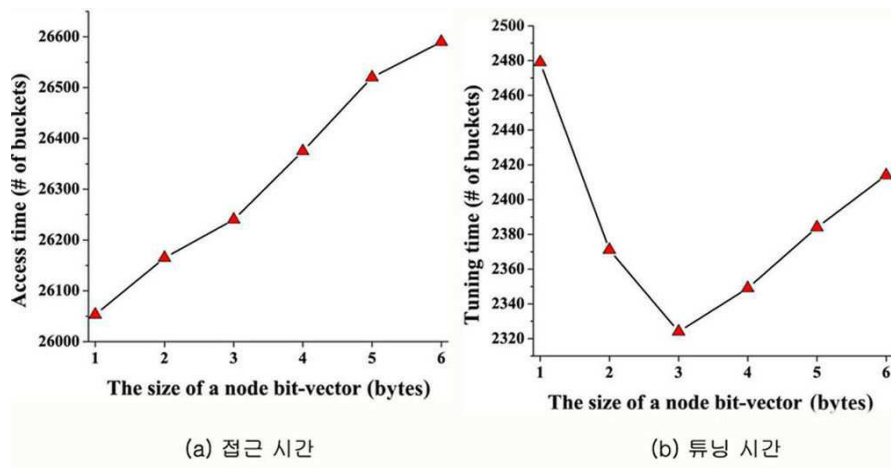
도면6



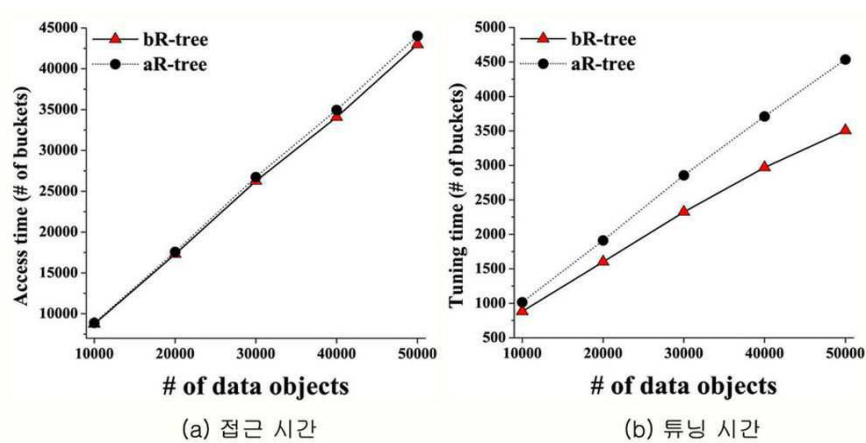
도면7



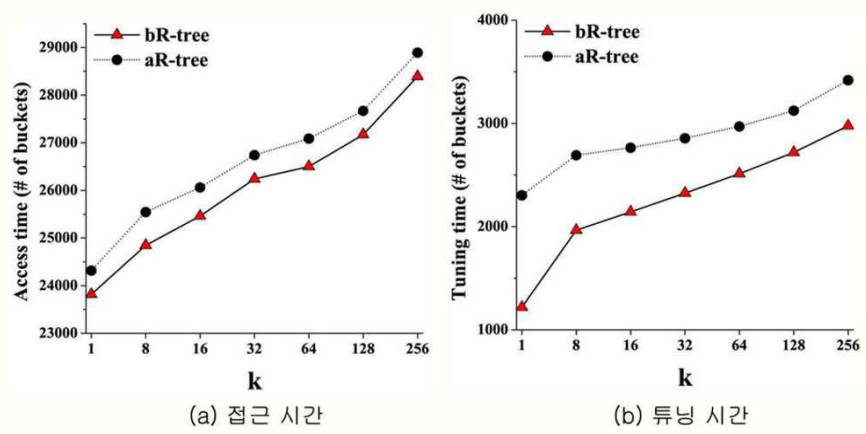
도면8



도면9



도면10



도면11

