편향 접근 패턴을 갖는 공간 데이터에 대한 공간 색인 기법

이승중[O] 정성원
서강대학교 컴퓨터학과
neoleesj[O]@sogang.ac.kr, jungsung@ccs.sogang.ac.kr

A Spatial Indexing Scheme for Geographical Data with Skewed Access Patterns

Seung-Joong Lee[0] Sungwon Jung
Dept. of Computer Science, Sogang University

요 약

차량항법장치(Car Navigation System : CNS)나 지리정보시스템(Geographic Information System : GIS)에서 공간 객체를 효율적으로 다루는 색인기법에 대한 다양한 논의가 있어왔다. 기존의 방법에서는 공간 객체의 인접성(cluster)과 밀집성만을 고려해서 색인 트리를 생성하므로, 편향된 접근 빈도를 가진 공간 객체에 대해서 효과적인 탐색시간을 제공하지 못한다. 접근 빈도를 반영한 색인 기법은 공간 데이터가 갖는 특성-2개 이상의 차원에 대한 순서 할당이 불가능-에 의해서 지리적으로 인접된 객체들을 묶지 못하고, 이로 인해서 공간 객체에 대한 효율적인 색인 기법을 제공할 수 없다. 지리 데이터에 대한 위치와 접근 빈도가 주어질 때, 색인 트리는 좌표 정보뿐 아니라 공간 객체에 대한 접근 빈도도 고려해서 생성되어야 한다. 본 논문에서 제안하는 기법은 전체 영역을 세부영역으로 분할하고, 각 세부 영역에 대해서 편향색인 트리를 생성한 뒤에 트리를 병합함으로써 밀집도와 접근 빈도를 반영한, 편향된(skewed) 색인 트리를 생성하도록 한다. 편향된 색인 트리는 접근 빈도가 높은 공간객체를 상위계층(level)에 위치시킴으로써 탐색비용을 줄인다.

1. 서 론

GPS및 PDA의 발달로 인해서 위치 기반 서비스(LBS), 차량항법장치(CNS), 지리정보시스템(GIS)등 공간 데이터를 다루는 응용프로그램들이 급속하게 보급되었다[1,2]. 응용프로그램은 사용자가 요청한 공간 객체와, 공간 객체의 주변에 위치한 공간 객체들을 사용자에게 보여준다. 공간 객체들은 접근 빈도를 가지고 있으므로, 접근 빈도에 따라서 편향식(skewed) 색인 트리를 구성하는 것이 접근 빈도가 높은 공간 객체에 대한 탐색시간을 줄인다. 지리정보시스템 내에서 주유소의 접근빈도는 학교의 접근빈도보다 높다고 가정을 하면, 색인 트리 내에서 주유소 객체의 위치를 학교 객체의 위치보다 상위에 배치시킴으로써, 사용자들에게 주유소 객체에 대한 빠른 탐색시간을 제공한다.

최근에는 1차원 객체들의 빈도를 반영해서 편향된(skewed) 색인 트리를 구성함으로써, 접근 빈도가 높은 객체에 대해서 빠른 탐색시간을 제공하는 기법이 연구되고 있다[3]. 2차원 이상의 객체에 대한 색인 트리에서는 공간 객체간의 지리적 인접성을 반영해서 색인을 생성해야만 효과적인 탐색이 가능하다. R-tree기반의 색인 기법은 인접한 공간 객체들을 묶어서 색인을 생성하므로, 공간 객체들의 지리적 인접성을 반영한다. 또한 색인 트리 내에서 공간 객체들의 깊이(depth)를 동일하게 해 줌으로써, 모든 공간 객체들이 동일한 접근 빈도를 갖는 경우에는 효율적인 접근 방법을 제공한다. 그러나, 모두 다른 접근 빈도를 갖는 공간 객체에 대해서도, R-tree기반의 색인 기법은 동일한 깊이(depth)를 제공한다. 그러므로, 사용자들의 요구가 많아지는 공간 객체에 대해서도, 빈도수가 낮은 공간 객체와 동일한 탐색시간을 제공하는 문제점이 발생한다.

빈도수만을 고려한 공간 객체의 색인 방법은, 접근 빈도에 따른 편향성(skewed)은 제공하지만, 지리적으로 밀집되어 있는 공간 객체에 대해서 효율적인 색인을 제공하기 어렵다. 이는 공간 객체가 2개 이상의 차원에 대한 정보를 필요로 하므로, 하나의 차원에 대해서만 정렬을 할 수 없기 때문에 발생하는 문제점이다. 공간 객체에 대해서 색인을 제공하면서, 동시에 접근 빈도에 따른 편향성을 제공함으로써, 접근 빈도가 높은 공간 객체에 대한 탐색시간을 줄이는 공간 색인 기법에 대한 연구가 필요하다.

본 논문에서는 밀집되어 있는 공간 객체의 접근 빈도를 반영해서 편향된 색인 트리를 생성하는 기법을 제안한다. 색인 트리 상에서 공간 객체의 깊이(depth)를 접근 빈도에 따라 다르게 할당함으로써, 접근 빈도가 높은 공간 객체에게 빠른 탐색시간을 제공하도록 한다.

본 논문의 구성은 다음과 같다. 제 2장에서는 관련 연구와 문제점에 대해서 살펴보고, 3장에서는 본 논문에서 제안하는 편향 색인 트리 생성 기법을 소개하며, 4장에서는 결론 및 향후 과제에 대해서 논의 한다.

2. 관련연구

데이터 접근 빈도를 반영한 색인 트리 생성방법은 호프만 트리와 Alphabetic 호프만 트리에서 적용되었다[3]. Alphabetic 호프만 트리는 하나의 차원에 대해서 순서를 갖는 객체의 구분자를 정렬한 다음, 인접된 객체의 접근 빈도 합이 가장 작은 객체들을 병합해서 색인 트리를 만드는 기법이다. Alphabetic 호프만 트리를 이용한 색인트리 생성 방법은, 객체들의 접근 빈도를 효율적으로 반영하여, 편향된 트리를 생성한다[3,4]. 그러나, Alphabetic 호프만 트리는 하나의 차원에 대해서 순서를 갖는 객체에 대해서만 편향성을 제공하므로, 2개 이상의 차원을 고려해야 하는 공간 객체에는 부적합하다.

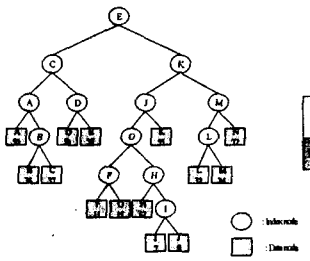


그림 1 Alphabetic 호프만 트리

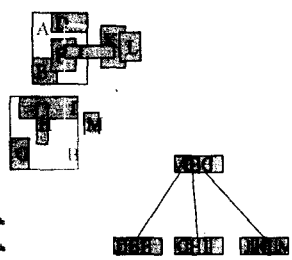


그림 2 R-tree

R-tree기반의 색인방법은 공간 객체간의 최단거리를 갖는 공간 객체를 묶어서 최소경계사각형을 생성해 나가는 기법이다. 최소경계사각형을 또 다른 공간 객체로 인식하여, 기본영역을 포함하는 최소경계사각형이 하나가 될 때까지 생성하는 방법이다. 이 방법은, 공간 객체의 거리상 근접성을 이용해서 트리를 생성하는 방법이다.[5]

R-tree를 이용한 색인 기법은 적은 계층으로 많은 수의 공간 객체를 색인할 수 있으므로 검색을 수행할 때, 탐색해야 하는 색인 노드(node)

의 수가 적다는 장점이 있으나, 공간 객체의 접근 빈도를 반영하지 못하는 단점을 가지고 있다.

본 논문에서는, 공간 객체에 대해서 지리적 밀집도와 접근 빈도를 반영한 색인 방법을 제안한다. 빈도수가 높은 공간 객체는 색인 트리 상에서 상위계층에 존재하고, 빈도수가 낮은 객체는 색인 트리에서 하위 계층에 존재함으로써, 자주 접근 되는 공간 객체에 대해서 보다 빠른 탐색시간을 제공한다.

3. 편향 색인 트리 생성 기법

본 논문에서는 편향 색인 트리를 생성하는 기법을 2단계로 구분해서 설명한다. 1단계에서는 전체 영역에 분포되어 있는 공간 객체들을 지리적 밀집도에 따라서 구분하기 위해서 하향식 방법으로 세부 영역을 묶는다. 1단계를 통해서 묶인 각 세부 영역에 대해서, 2단계에서는 상향식 방식으로 접근 빈도를 반영해서 편향 색인 트리를 생성한다. 접근 빈도가 높은 공간 객체가 접근 빈도가 낮은 공간 객체보다 색인 트리 상에서 더 깊은 깊이(depth)를 가지는 문제점을 해결하기 위해서, 각 세부영역간의 접근 빈도를 비교해서 세부영역을 하나의 공간 객체로 인식하고, 이를 포함해서 접근 빈도를 적용해서 편향 색인 트리를 생성한다.

3.1 전체영역을 지리적 밀집도에 따라서 세부영역으로 묶기

편향 색인 트리 생성 기법 1단계에서는 전체 공간영역에서, 인접된 공간 객체들끼리 동일 영역으로 묶어줌(clustering)으로써, 공간 객체들 간에 지리적으로 인접한 특성을 유지해준다. 또한, 1단계 기법을 이용해서 계산 영역을 제한함으로써, 2단계 기법의 계산 비용을 줄여준다.

1단계 기법에서 사용하는 용어에 대해서 정리한다.

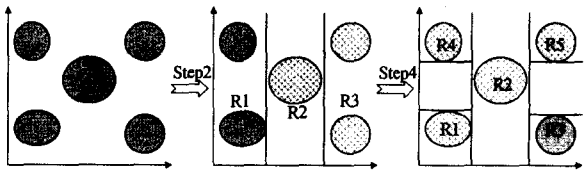
- R : 공간객체들이 나열되어 있는 전체 영역
- R_i : 1단계 기법에 의해서 분할된 세부 영역
- OBJ_i : i번째 공간 객체
- $AveDX$: X축에서 공간 객체간의 평균거리
- $AveDY$: Y축에서 공간 객체간의 평균거리
- DX_{ij} : X축 상에서 공간객체 i와 j사이의 거리
- DY_{ij} : Y축 상에서 공간객체 i와 j사이의 거리

step1 : R에 속하는 모든 공간 객체를 공간객체의 X좌표에 대해서 올림차순으로 정렬한다.
step2 : X좌표에 대해서 정렬된 모든 공간객체에 대해서,
if ($DX_{i+1} < AveDX$)
    $R_j = R_j \cup \{OBJ_i, OBJ_{i+1}\}$
else if ($DX_{i+1} \geq AveDX$) {
    $R_j = R_j \cup \{OBJ_i\}$
    $R_{j+1} = \{OBJ_{i+1}\}$
    $j = j + 1$
}
step3 : step2에서 생성된 모든 세부영역에 대해서,
각각의 세부영역에 포함되는 모든 공간 객체들을 공간객체의 y좌표에 대해서 올림차순으로 정렬한다.
step4: step2에서 생성된 세부영역의 수를 M이라 하면,
for ($R_j = R_1$; $R_j \leq R_{M+K}$; 현재 세부영역의 다음 세부영역) {
  for (세부영역 R_j에 속하는 모든 공간객체OBJ_i) {
    if ($DY_{i+1} \geq AveDY$) {
      R_j에 포함된 $i+1$번째 이후의 공간객체를 R_j에서 삭제하고, 세부영역 R_{M+K}에 포함시킨다.
      $K = K + 1$
    }
  }
}

그림 3. 전체 영역을 세부 영역으로 분할하는 편향 색인 트리 생성 기법

그림3은 전체 영역을 세부 영역으로 분할하는 1단계 기법을 서술한다. step2는 전체 영역 R을 X축에 대해서 세부 영역으로 분할한다. step3는, step2에서 구분된 세부 영역들에 대해서, y축으로 추가 분할함으로써, 밀집되어 있는 공간 객체들이 같은 세부 영역에 위치시킨다.

그림 4는 1단계 기법을 적용해서 전체 영역을 세부영역으로 나누는 과정을 도식화하였다. step4에서는 step2에서 분할되었던 세부영역과, step4를 수행하면서 추가로 생성된 세부영역에 대해서 영역분할을 수행하게 된다. 그러므로, 그림 4(c)에서와 같이 공간객체들의 밀집도에 따라서 효과적으로 세부영역을 나눌 수 있다.



(a) (b) (c)

그림 4. 전체 영역에 대해서 1단계 기법을 적용시켜서 세부영역을 분할한 개념도

그림 5는 그림3에 의해서 전체 영역 R이 세부영역 R1과 R2로 분할되는 것을 설명한다. 이 예제에서는 그림3의 step2에 의해서 전체영역 R이 세부 영역 R1과 R2로 분할되지만, 세부 영역 R1과 R2에는 step4에 의해서 분할되지는 않는다.

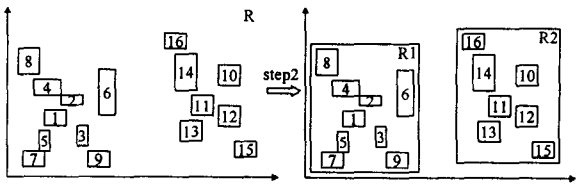


그림 5. 전체 영역R을 step2에 의해서 세부영역 R1과 R2로 분할

3.2 세부 영역에 대해서 접근빈도를 적용

거리상 인접한 공간객체들 중에서 접근 빈도가 가장 낮은 객체들을 병합하는 방법이 2단계 기법이다. 2단계 기법을 통해서, 각각의 공간 객체의 접근빈도가 반영되며, 색인트리가 편향되는 구조를 갖게 된다. 이러한 편향구조의 색인트리는 접근 빈도가 높은 공간객체에 대해서 짧은 검색시간을 제공해준다.

step1에서 세부 영역 정렬을 위한 기준으로써 다음을 사용한다.

- $Density_i$: 세부 영역 R_i내에 속하는 공간객체들의 평균 접근 빈도

step1: 세부 영역의 접근빈도에 대해서 올림차순으로 정렬
$Density_i$에 대해서 각 영역을 올림차순으로 정렬한다. 정렬 이후에 가장 작은$Density_i$를 가지고 있는 세부영역부터 step2이후의 과정을 진행한다.

step2: 접근 빈도가 작은 세부 영역 병합
해당 영역에 포함된 공간 객체의 최소 접근 빈도가, x축 또는 y축으로 인접한 세부 영역에 속한 공간객체의 최대 접근 빈도보다 클 경우, 인접 영역을 해당 영역에 속한 하나의 공간객체로 간주하여 해당영역에 삽입한다. 공간객체화 된 세부영역은 삭제한다.

step3: 각 공간객체의 인접 객체 선택
해당 영역의 각 공간 객체에 대하여, 거리상으로 가장 인접한 공간객체를 찾아서 거리가 가까운 순으로 순위를 부여한다.

step4: 인접객체 간의 빈도합 계산
step3에서 계산한, 공간객체와 거리상 가장 가까운 인접 공간객체의 접근빈도 합을 계산한 뒤, 접근빈도 합에 대해서 올림차순으로 정렬한 뒤 순위를 부여한다.

step5: 거리 순위와 빈도순위에 의한 병합 객체 선택
step3과 step4에서 부여된 순위를 더한 뒤에 순위의 합이 가장 작은 두 공간 객체를 최소경계사각형으로 묶는다. 이 때 새로 생긴 최소

영역사각형의 접근 빈도는 묶인 두 객체의 접근빈도의 합과 같다.
step6: 종료조건 판별
해당 영역에서 공간 객체의 수가 한개만 남으면 step7로 이동하고, 그렇지 않을 경우 step3로 이동한다.
step7: 다음세부영역 선택
정렬된 세부 영역 순열에서, 다음 세부 영역이 존재하면, 다음 세부 영역을 선택한 뒤 step2로 이동한다. 다음 세부 영역이 없을 경우 종료한다.

그림 6. 세부영역에 대해서 빈도를 반영하기 위한 색인 트리 생성 기법

그림 6의 step2는 1단계 기법을 적용할 때, 발생하는 문제점을 해결한다. 전체영역에서는 접근 빈도가 높음에도 불구하고, 1단계 기법을 적용시킴으로써, 접근 빈도가 낮은 공간 객체가 전체 트리의 상위계층에 존재하는 문제점이 발생하므로, 이러한 문제를 해결하기 위해서 step2를 적용해서 접근 빈도가 높은 공간 객체가 색인 트리에서 상위 계층에 존재하도록 tree를 조정한다.

2단계 기법을 통해서 생성된 트리는 step3과 step4에 의해서 객체의 빈도와 지리적 인접성을 반영하게 된다. 지리적 인접 순위와 접근 빈도 순위의 합을 이용하므로, 빈도수가 높은 공간 객체는 상대적으로 빈도수가 낮은 공간 객체보다 늦게 묶이게(Bound) 된다. 늦게 묶인 공간 객체는 편향 색인 트리 상에서 상위 계층에 위치하게 되므로, 상향식 알고리즘에 의해서 접근 빈도를 포함한 색인 트리가 완성된다.

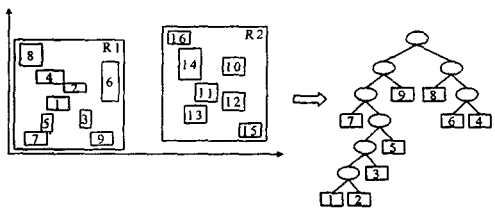


그림 7. 세부 영역 R1에 대한 편향 색인 트리 생성

그림 7은 세부 영역에 대해서 접근 빈도를 적용한 편향 색인 트리를 생성하는 과정을 설명한 그림이다. 3.1에서 설명한 1단계 기법에 의해서 전체 영역을 세부 영역 R1과 R2로 분할한다. 분할된 세부 영역들에 대해서 $Density_1$과 $Density_2$의 값에 대해서 올림차순으로 정렬한다. 먼저 세부 영역 R1에 대해서, 2단계 기법을 적용한다. 세부 영역 R1이 $Density_1$의 값이 가장 작으므로 step2에 의해서 선택되는 주변의 세부 영역은 존재하지 않는다. step3~step6의 과정을 적용시켜서 세부 영역 R1에 속하는 공간 객체에 대한 편향 색인 트리를 생성한다.

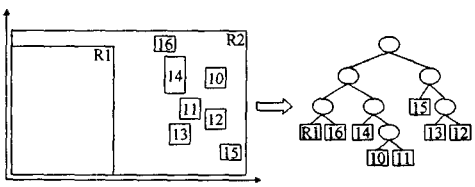


그림 8. 세부 영역 R2에 대한 편향 색인 트리 생성

그림 8은 세부 영역 R2에 대한 편향 색인 트리를 생성한다. step2에 의해서 접근 빈도가 낮은, 인접한 세부 영역 R1을 하나의 공간 객체로 포함하며, 기존의 세부 영역의 범위를 공간 객체 R1을 포함하는 영역으로 확장시킨다. 공간 객체 R1을 포함해서 확장된 세부 영역 R2에 대해서, step3~step6의 과정을 통해서 편향 색인 트리를 생성한다. 그림 9는 세부 영역 R2에 대해서 완성된 편향 색인 트리에 그림 7에서 생성된 편향 색인 트리를 공간 객체 R1의 위치에 붙임으로써 전체 영역에 대한 편향 색인 트리는 완성되는 모습을 도식화시켰다.

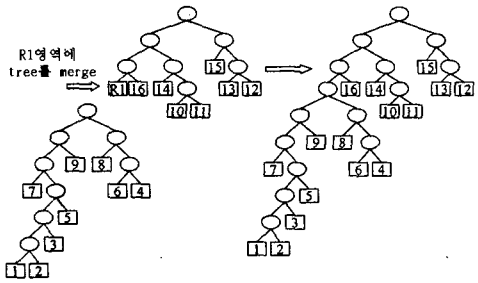


그림 9. 세부 영역 R1에 대한 색인 트리를 세부 영역 R2에 대한 편향트리에 붙인 후 완성된 편향 색인 트리

4. 결론

기존에 연구된 공간 객체에 대한 색인 생성 방법은, 공간 객체에 대한 접근 빈도를 고려하지 않았기 때문에, 빈도수에 상관없이 동일한 탐색시간을 제공하였다. 본 논문에서는, 공간 객체들의 지리적인 인접성과 접근 빈도를 반영한 편향 색인 트리를 생성하는 기법을 제안하였다. 공간 객체에 대해서 접근 빈도를 반영한 편향 트리를 생성함으로써, 편향 색인 트리 내에서 빈도수가 높은 공간 객체에 대한 적은 탐색시간을 제공한다. 전체 영역을 세부 영역으로 구분하고 세부 영역에 대해서 편향 색인 트리를 생성함으로써 발생하는 문제점-전체영역에서는 접근 빈도가 높은 공간 객체가 세부 영역을 구분해서 트리를 생성하기 때문에 색인 트리 생성 후 접근 빈도가 낮은 공간 객체보다 하위계층에 위치-을 세부영역간의 접근 빈도 비교를 통해서 조정하였다. 트리를 생성하기 위해서 전체 공간 객체와 세부 영역에 대한 검색이 필요하므로, 전체 트리를 생성하는데 많은 계산 비용이 드는 단점이 있다. 또한, 각 노드의 fanout을 증가시킬수록 접근 빈도가 높은 공간 객체와 낮은 공간 객체간의 단계(level)차이가 적게 난다. 그러므로, fanout의 크기와 색인 트리 상에서의 단계차이와의 상관관계를 고려해서 적절한 fanout을 결정하는 것이 트리의 성능을 결정한다.

5. 참고문헌

[1] Zhang, J., Zhu, M., Papadias, D., Tao, Y., Lee, D. "Location-Based Spatial Queries. To appear in Proceedings of ACM Conference on Management of Data" SIGMOD, pp 467-478, June 9-12, 2003

[2] Ayse Y. Seydim, Margaret H. Dunham, Vijay Kumar "An Architecture for Location Dependent Query Processing" MDDS 01, DEXA Workshop, 2001

[3] Narayanan Shivakumar and Suresh Vnkatasubramanian, "Efficient indexing for broadcast based wireless systems", Mobile Networks and Applications, Vol. 1, pp. 433-446, 1996.

[4] D. Knuth, "The Art of Computer Programming Second Edition, Vol III", Addison Wesley, 1998

[5] Antonin Guttman, "R-Trees : A Dynamic Index Structure for Spatial Searching", Proceeding of the 1984 ACM SIGMOD International Conference on Management of data June 1984