RAH-tree : 편향 접근 패턴을 갖는 공간 데이터에 대한 효율적인 색인 기법

최근하° 이승중 정성원
서강대학교 컴퓨터학과
purymul°@sogang.ac.kr, neoleesj@sogang.ac.kr, jungsung@ccs.sogang.ac.kr

RAH-tree : A Efficient Index Scheme for Spatial Data with Skewed Access Patterns

Keun-Ha Choi° Seung-Joong Lee Sungwon Jung
Dept. of Computer Science, Sogang University

요 약

GPS및 PDA의 발달로 인해서 위치 기반 서비스(LBS), 차량항법장치(CNS), 지리정보시스템(GIS)등 공간 데이터를 다루는 응용프로그램들이 급속하게 보급되었다. 이러한 응용프로그램은 높이 균등 색인 기법을 사용하여 원하는 데이터에 대한 색인을 제공하였다. 그러나 모든 공간 객체는 서로 상이한 접근 빈도를 가지고 있음에도 불구하고 기존의 공간 색인 기법은 접근 빈도를 고려하지 못하는 단점을 가지고 있었다. 또한 기존의 빈도수만을 고려한 공간 객체의 색인 방법은 접근 빈도에 따른 편향성(skewed)은 제공하지만 공간 객체에 대한 지역성을 반영하지 못한다. 본 논문에서는 밀집되어 있는 공간 객체의 접근 빈도를 반영해서 편향된 색인 트리를 생성하는 기법을 제안한다. 이형 클러스터링으로 분포되어 있는 전체 영역에 대해서 Zahn의 클러스터링 알고리즘을 변형시켜서 다단계 세부영역을 구분한다. 이렇게 구분된 세부영역에 대해서 거리적 인접성과 접근 빈도수의 합을 이용해서 색인 트리를 생성한다. 다단계로 구성된 전체 영역에 대해서 하향식 방식으로 편향된 색인 트리를 생성함으로써, 접근 빈도가 높은 공간 객체에 대해서 빠른 탐색이 가능하게 한다.

1. 서 론

모바일 컴퓨팅 환경이 보급되면서 텔레매틱스, GPS및 PDA의 사용자가 증가하고 위치 기반 서비스(LBS), 차량항법장치(CNS), 지리정보시스템(GIS)등 공간 데이터를 다루는 응용프로그램들이 급속하게 보급되었다[1,2]. 이를 위해서 지금까지는 지리 정보와 같은 공간 데이터에 대한 색인 기법으로 R-tree기반의 높이 균등 트리기법에 기반을 둔 색인 기법이 사용되었다. 이러한 색인 기법들은 모든 공간 객체에 대해서 동일한 높이를 제공하여 사용자가 원하는 정보를 탐색할 수 있도록 하였다. 하지만 공간 객체들의 접근 빈도에 따라서 색인 트리의 레벨(level)을 달리하여 이동 단말기들이 자주 요구하는 공간 객체에 대한 탐색시간을 줄이고 보다 빠르게 원하는 정보를 전송할 수 있다.

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. 하지만 서로 다른 접근 빈도를 갖는 공간 객체에 대해서도 동일한 레벨(level)을 제공하기 때문에 사용자들의 요구가 많아지는 공간 객체에 대해서도, 빈도수가 낮은 공간 객체와 동일한 탐색시간을 제공하는 문제점이 발생한다. 반면에 빈도수만을 고려한 공간 객체의 색인 방법은 접근 빈도에 따른 편향성(skewed)은 제공하지만 지리적으로 밀집되어 있는 공간 객체에 대해서 효율적인 색인을 제공하기 어렵다. 그러므로 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.

이 논문에서는 밀집되어 있는 공간 객체의 접근 빈도를 반영해서 편향된 색인 트리를 생성하는 기법인 RAH-tree (Rectangular Alphabetic Huffman tree)를 제안한다. 이 기법은 R-tree의 공간 색인 기법과 Alphabetic Huffman tree의 편향성을 결합하여 접근 빈도가 높은 공간 객체에 대해서 효율적인 색인 정보를 제공한다.

본 논문의 구성은 다음과 같다. 제 2장에서는 관련 연구와 문제점에 대해서 살펴보고, 3장에서는 본 논문에서 제안하는 편향 색인 트리 생성 기법에 대해서 소개하며, 4장에서는 실험을 통해 본 논문에서 제안하는 알고리즘과 기존 알고리즘과의 성능을 비교 및 분석하여 제안한 알고리즘을 평가한다. 마지막으로 5장에서는 결론과 향후 연구 방향을 기술한다.

2. 관련연구

본 논문에서 제안하는 공간색인 기법에 대해서 설명하기 전에 Alphabetic Huffman tree와 R-tree에 대해서 알아보자.

Alphabetic Huffman tree는 트리의 상위 부분에 접근 빈도가 높은 데이터가 위치시킴으로써 상대적으로 높은 접근 빈도를 갖는 데이터에 대해서 작은 탐색시간(search time)을 갖도록 한다[3,4]. 그러나 Alphabetic Huffman tree는 하나의 차원에 대해서 순서를 갖는 객체에 대해서만 편향성을 제공하지만 2개 이상의 차원을 갖는 공간 객체에는 부적합하다.

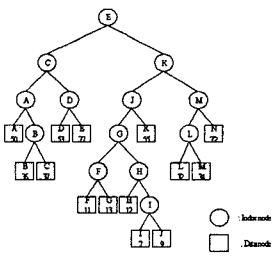

그림 1 Alphabetic 호프만 트리

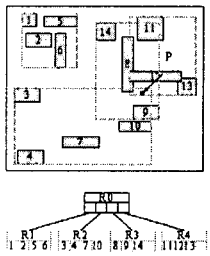

그림 2 R-tree

R-tree는 공간 객체간의 지리적 인접성을 기반으로 공간 객체를 묶어서 최소경계사각형을 생성해 나가는 기법이다[5]. 최소경계사각형을 또 다른 공간 객체로 인식하여 기본영역을 포함하는 최소경계사각형이 하나가 될 때까지 반복한다. R-tree는 적은 계층으로 많은 수의 공간 객체를 색인함으로써 탐색해야 하는 색인 노드(node)의 수가 적다는 장점이 있으나 공간 객체의 접근 빈도를 반영하지 못하는 단점을 가지

고 있다. 본 논문에서는 Alphabetic Huffman tree와 R-tree의 특징을 통하여 공간 객체에 대해서 접근 빈도를 반영한 RAH-tree (Rectangular Alphabetic Huffman tree)를 제안한다.

3. RAH-tree

본 논문에서는 편향 색인 트리를 생성하기 위해 하향식 접근법과 상향식 접근법, 2단계로 구분해서 설명한다. 1단계에서는 하향식 접근법으로 전체 영역에 나열된 공간 객체들의 클러스터링 영역을 찾아서 세부영역으로 나눈다. 2단계에서는 각 세부영역에 대해서 상향식 접근법으로 거리적 인접성과 접근빈도를 사용하여 접근 빈도가 높은 데이터를 트리의 상위 레벨에 위치시킴으로써 편향된 색인 트리를 생성한다.

3.1 전체 영역 클러스터링

전체 영역에 나열되어 있는 공간 객체를 여러 개의 세부영역으로 구분하기 위해서 Zahn의 클러스터링 알고리즘을 응용하였다. RAH-tree 생성 기법에서 클러스터링 기법은 공간 객체들이 속해 있는 공간을 작은 세부영역으로 구분함으로써 3.2 장에서 설명할 상향식 기법의 계산양을 줄여주는 효과를 얻는다. Zahn의 clustering 알고리즘을 설명하기 전에 inconsistent 간선에 대해서 정의한다.

정의 1. 최소신장트리의 간선 중에 다음의 조건을 만족하는 간선 e_i을 inconsistent 간선이라고 한다. 만약 간선 e_i가 자신을 제외한 나머지 간선들의 평균 거리보다 큰 유클리드 거리를 갖는 경우에 간선 e_i는 inconsistent 간선이라고 한다.

정의 1에서 inconsistent 간선을 현재 간선을 제외한 나머지 간선들의 평균거리와 비교하는 이유는 만약 현재 간선이 inconsistent 간선이라면 평균값을 크게 함으로써 inconsistent 간선을 발견하지 못하는 문제가 발생할 수 있기 때문이다. Zahn의 클러스터링 알고리즘은 입력으로 들어온 Graph에 대해서 최소 신장 트리를 생성한다. 생성된 최소 신장 트리에 대해서 inconsistent 간선을 제거하고 남은 연결성을 가지는 node의 집합을 세부영역으로 간주한다[6,7]. 그림 2는 RAH-tree의 클러스터링 알고리즘을 보여준다.

OBJECTS = {obj_1, obj_2, ..., obj_m}, obj_i는 공간객체
REGION = {R_1, R_2, ..., R_n}, R_i는 분할된 세부영역
INPUT : *OBJECTS*
OUTPUT : *REGION*
BEGIN
STEP 1: 모든 obj_i의 중점 좌표에 대해서 최소신장트리를 생성한다.
STEP 2: 최소신장트리의 간선 중 inconsistent 간선을 표시한다.
STEP 3: inconsistent 간선이 존재하지 않으면 종료한다.
STEP 4: STEP 2에서 표시된 inconsistent 간선을 제거하고 연결된 요소들을 세부영역 R_j로 구성한다.
STEP5. STEP 4에 의해서 생성된 각각의 R_j에 대해서 STEP 1을 반복한다.
END

그림 3 RAH-tree의 클러스터링 알고리즘

RAH-tree의 클러스터링 알고리즘은 기본적으로 Zahn의 방법과 동일하지만 Zahn과는 달리 다단계 클러스터링 검색을 실시함으로써 접근빈도를 효과적으로 반영할 수 있도록 한다. 예를 들어 클러스터링 되어 있는 영역에 접근 빈도가 높은 공간 객체가 존재함에도 불구하고 공간 객체의 수가 밀집되어 있는 경우 전체 RAH-tree상에서 상위 영역에 존재할 수 없는 문제점이 발생한다. 이런 경우 STEP 4에서 생성된 세부영역에 대해서 클러스터링을 재실행함으로써 빈도수가 높은 공간 객체가 보다 상위에 위치할 수 있도록 한다.

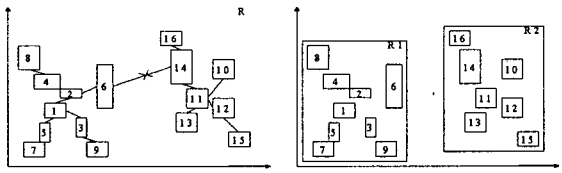

그림 4 RAH-tree를 구성하기 위한 클러스터링 기법

그림 4는 RAH-tree를 구성하기 위한 클러스터링 알고리즘의 동작 예를 보여준다. STEP 2에서 최소신장트리의 간선 중 inconsistent 간선(x 표시된 간선)을 조사하고 이를 제거하여 전체 영역을 R1과 R2로 나눈다. 각 세부영역은 더 이상의 inconsistent간선이 존재하지 않으므로 STEP 4에서 종료된다.

3.2 세부영역에 대해서 RAH-tree 생성

이 장에서는 공간 객체들의 거리적 인접성과 접근빈도를 반영해서 편향된 색인 트리를 생성하는 기법에 대해서 설명한다. 이러한 편향구조의 색인트리는 접근 빈도가 높은 공간객체에 대해서 짧은 검색시간을 제공해준다. 그림 5는 그림 3에 의해서 분할된 세부영역을 통해서 RAH-tree를 생성하는 알고리즘을 보여준다.

CenterCord(obj_i)는 obj_i의 중점을 의미한다.
$DISTANCE(obj_1, obj_2) = \sqrt{((CenterCord(obj_1) - CenterCord(obj_2))^2}$
REGION = {R_1, R_2, ..., R_n}
R_i = {obj_1, obj_2, ..., obj_{di} },
R_i 는 그림 3의 클러스터링 알고리즘을 통해 분할된 세부영역
$Density_i = \sum_{i=1}^{d_i} freq_i / d_i$, $freq_i$는 R_i의 공간객체 obj_i의 접근 빈도
INPUT : *REGION*
OUTPUT : *RAH-tree*
BEGIN
R_i의 평균접근빈도 $Density_i$에 대해서 각 영역을 오름차순으로 정렬하고 가장 작은 *Density*를 가지고 있는 세부영역부터 다음을 과정을 반복한다.
STEP 1: *Density*가 작은 R_{i-1} 병합
R_i보다 *Density*가 낮은 R_{i-1}을 R_i의 공간객체로 간주하고 R_i의 영역이 R_{i-1}을 포함하도록 확장한다.
STEP 2: 각 공간 객체의 인접 객체 선택
 while (R_i의 obj_k에 대해서) {
  while (R_i에 포함된 나머지 obj_l에 대해서) {
   obj_k와 obj_l간의 유클리드거리(*DISTANCE*(obj_k,obj_l))를 계산
   obj_k와 가장 인접한 객체정보를 수정한다.
  }
 }
 R_i의 공간객체에 대해서 계산된 인접 객체와의 거리 정보를 기준으로 오름차순으로 정렬 후 순위를 부여한다.
STEP 3: 인접 객체 간의 빈도 합 계산
 while (R_i의 obj_k에 대해서) {
  while (R_i에 포함된 나머지 obj_l에 대해서) {
   obj_k와 obj_l간의 접근빈도의 합을 구한다.
   obj_k와 가장 작은 접근빈도 합을 갖는 객체정보를 수정한다.
  }
 }
 R_i의 전체 객체에 대하여 인접 객체와의 접근 빈도 합을 기준으로 오름차순으로 정렬 후 순위를 부여한다.
STEP 4: 거리 순위와 빈도 순위에 의한 객체 선택 및 병합
 1. R_i에 대하여 STEP2와 STEP3에서 부여된 순위를 더한 후에 최종 순위의 합이 가장 작은 두 공간객체를 선택한다.
 2. 선택한 두 공간 객체를 최소경계사각형으로 병합한다. 만약 순위의 합이 동일한 경우 빈도 합이 작은 쪽을 선택한다. 새로 생긴 최소경계사각형의 접근 빈도는 병합된 두 객체의 접근 빈도의 합과 같다.
 4. 병합된 최소경계사각형을 R_i의 새로운 공간 객체로 포함시킨다.
STEP 5: 종료조건 판별
 R_i의 공간 객체의 수가 한 개가 남을 때까지 STEP1~STEP4를 반복한다.

그림 5 RAH-tree 생성 알고리즘

RAH-tree 생성 알고리즘을 통해서 생성된 트리는 STEP 2와 STEP 3에 의해서 객체의 빈도와 지리적 인접성을 반영하게 된다. 이런 특징으로 인해 빈도수가 높은 공간 객체는 상대적으로 빈도수가 낮은 공간 객체보다 늦게 묶이게(Bound) 되고, 편향 색인 트리 상에서 상위 계층

에 위치하게 되어 상향식 알고리즘에 의해서 접근 빈도를 포함한 색인 트리가 완성된다.

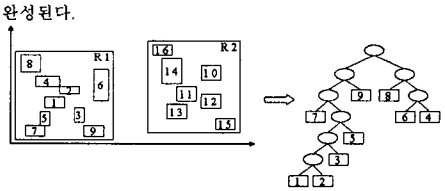

그림 6 세부영역 R1에 대한 편향 색인 트리 생성

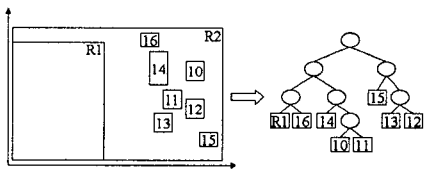

그림 7 세부영역 R2에 대한 편향 색인 트리 생성

그림 6은 3.1장에서 설명한 클러스터링 알고리즘에 의해 생성된 세부영역 R1과 R2에 대해서 편향 색인 트리를 생성하는 과정을 설명한 그림이다. 세부영역 R1과 R2의 공간객체에 표시된 접근빈도에 따라 R1의 *Density*는 5이고, R2의 *Density*는 13이 된다. *Density*를 기준으로 *Density*가 작은 R1에 대한 편향 색인 트리를 우선적으로 생성하게 된다. R1에 대해서는 R1보다 *Density*가 더 작은 세부영역이 존재하지 않으므로 STEP 1은 수행하지 않는다. 그 이후에 STEP 2~STEP 3에 의해 병합할 세부영역의 공간 객체들을 선택하여 차례로 병합하는 과정을 반복한다.

그림 7은 세부영역 R2에 대한 편향 색인 트리를 생성하는 그림이다. STEP 1에 의해서 *Density*가 낮은 인접한 세부영역 R1을 R2의 공간 객체로 포함하며 기존의 세부영역의 범위를 확장시킨다. 확장된 세부영역 R2에 대해서 STEP 2~STEP 5의 과정을 통해서 그림 7의 편향 색인 트리를 생성한다. R2의 편향 색인 트리를 생성한 후에 그림 8에서와 같이 R1의 위치에 해당하는 편향 색인 트리를 병합함으로써 전체 영역에 대한 편향 색인 트리를 완성한다.

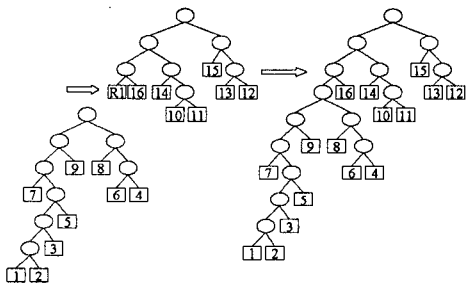

그림 8 세부영역 R1의 RAH-tree를 세부영역 R2에 병합

4. 성능분석

본 논문의 모든 실험은 다음과 같은 환경에서 수행되었다. 이 장에서는 제안하는 RAH-tree가 편향된 접근 패턴을 갖는 데이터에 효과적임을 증명한다. 또한 데이터의 양이 어떤 영향을 미치는가를 확인한다.

Machine	Sun Blade 1000
OS	Solaris 5.8
Processor	UltraSparc III
Memory	1 GB
Compiler	gcc Compiler

표 1 실험 환경

[그림 9]는 공간 객체에 대한 이동 단말들의 접근이 특정 객체에 집중 될수록 RAH-tree가 짧은 탐색 시간을 제공하는 것을 보여준다. 이는 접근 빈도가 높은 데이터가 트리의 상위 레벨에 존재하기 때문에 R-tree에 비해 Zipf factor가 1에 가까워질수록 실제 탐색 시간이 줄어드는 것을 볼 수 있다. [그림 10]은 데이터의 수가 증가하더라도 RAH-tree가 R-tree에 비해 보다 빠른 탐색시간을 제공하는 것을 알 수 있다. 이는 공간 객체의 수가 증가할수록 접근 빈도가 높은 공간 객체들과 상대적으로 접근 빈도가 낮은 공간 객체들 간의 트리의 레벨 차이가 커짐에 따라 접근 빈도가 높은 데이터에 대한 훨씬 빠른 탐색 시간을 제공하기 때문이다.

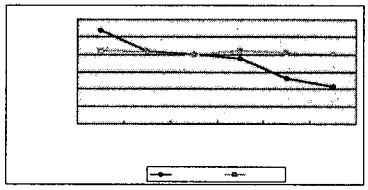

그림 9 접근 패턴에 따른 탐색시간

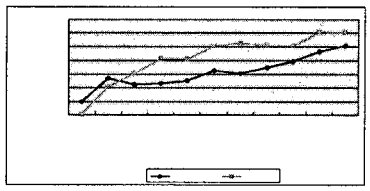

그림 10 공간 객체 개수에 따른 탐색시간

5. 결론

기존의 공간 객체에 대한 색인 생성 방법은 공간 객체에 대한 접근 빈도를 고려하지 않고 모든 공간 객체에 대해 동일한 탐색시간을 제공하였다. 본 논문에서는 기존의 R-tree와 Alphabetic Huffman tree의 특징을 반영하여, 공간 객체들의 지리적인 인접성과 접근 빈도를 반영한 RAH-tree를 제안하였다. RAH-tree는 이형 클러스터링 되어 있는 공간 객체들에 대해서 접근 빈도를 반영한 편향 트리를 생성하고 접근 빈도가 높은 공간 객체에 대해서 빠른 탐색시간을 제공하였다.

6. 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

[6] Anil K. Jain, Richard C. Dubes *Algorithms for Clustering Data*, Prentice Hall Advaced Reference series, pp. 120-128, 1988

[7] Zahn, C. T, *Graph-theoretical methods for detecting and describing Gestalt clusters*, IEEE transactions on Computers C20, pp. 68-86, 1971