밀도를 이용한 k-최근접 탐색 방법

장인성° 이기준
부 산 대 학 교 전 자 계 산 학 과
{isjang, lik}@quantos.cs.pusan.ac.kr

A Density-Based k-Nearest Neighbors Search Method

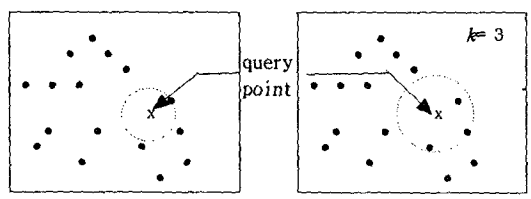
In-Sung Jang° Ki-Joune Li
Dept. of Computer Science, Pusan University

요 약

공간 데이터베이스 관리 시스템에서 제공하는 공간 질의는 많은 디스크 참조와 CPU 처리시간을 필요로 한다. 이 중에서 k-최근접 질의는 많은 디스크 참조를 요구하는 질의로써 지금까지 많은 연구가 이루어져 왔다. 트리 구조의 색인을 사용하는 k-최근접 질의 처리방법은, 조건을 만족하지 않는 노드를 가지치기 기법을 사용하여 노드 방문 횟수를 줄인다. 그러나, 이 방법은 가지치기과정에서 불필요한 디스크 참고가 발생하여 성능을 저하시키는 단점을 가지고 있다. 본 논문에서는 가지치기 기법 대신 주어진 k 개의 최근접 객체가 존재할 영역을 미리 예측함으로써 디스크 참조 횟수를 줄이는 방법을 제시한다. 이 영역을 예측하기 위해서 본 논문에서는 데이터 분포에 대한 밀도를 이용하였다. 실험에 의하면 이러한 방법은 기존의 가지치기 기법을 이용한 방법에 비해서 최고 22%, 평균 7%정도의 디스크 참조 횟수의 감소 효과가 있음을 알 수 있다.

1. 서론

공간 데이터베이스 관리시스템에서 다루는 공간 데이터는 그 양이 방대할 뿐만 아니라 형태가 복잡하여 질의 처리시 빈번한 디스크 입출력을 필요로 한다. 공간 데이터 베이스 관리 시스템에서 제공하는 질의는 점질의, 영역질의, 조인질의, 최근접 질의 등 매우 다양하다. 이 중에서 최근접 질의는 질의점에서 가장 가까운 공간 객체를 찾는 질의(그림 1-(a))로써 이것을 일반화시킨 것이 k-최근접 질의이다. 이는 주어진 질의점에서 가장 가까운 k개의 객체를 찾는 질의(그림 1-(b))로써 지금까지 이에 대한 연구는 많이 행해졌다.



(a) 최근접 질의 (b) 3-최근접 질의

그림 1. k-최근접 질의의 예

최근접 탐색을 수행하기 위해서는 모든 공간 객체와의 거리를 계산하여야 하는데, 방대한 공간 데이터를 다루는 공간 데이터베이스에서 이는 바람직하지 않다. 이를 고려하여, 지금까지 공간 색인 방법을 이용하여 불필요한 공간 객체는 비교하지 않는 여러 기법들이 소개되었다. 기존에 제안된 방법으로는 Cell-기반의 색인 방법인 Voronoi cell과 R*-tree[1]와 같은 트리-기반의 색인 방법인 MinMax 거리를 이용한 가지치기 방법[2][3], X-tree, SS-tree, SR-tree, Pyramid-tree와 클러스터링에 의한 방법등이 있다.

본 논문에서는, 트리-기반 공간 색인 기법을 이용하면서 객체의 분포 상태를 고려하여 k개의 최근접 객체가 존재할 영역을 미리 예측하고, 이를 질의 영역으로 이용하는 새로운 방법을 제안한다. 실험에 의하면, 본 논문에서 제안한 방법은 기존의 가지치기방법에 비하여 좋은 성능을 보여 주었다.

본 논문의 구성은 다음과 같다. 2장에서는 R*-tree를 이용한 가지치기방법에 대해서 소개하고, 3장에서는 본 논문에서 제시한 밀도기반 질의 영역 예측 방법을 이용한 k-최근접 질의 알고리듬을 제시한다. 4장에서는 본 논문에서 제시한 방법과 기존의 R*-트리를 이용한 가지치기 기법을 비교 실험한 결과를 제시하고, 마지막으로 5장에서는 결론과 향후 연구방향에 대해서 알아본다.

2. 가지치기(Pruning)방법을 이용한 k-최근접 탐색

본 장에서는 k-최근접 탐색을 위해 제안된 방법 중 R*-tree를 이용한 가지치기 방법[2][3]과 이 방법의 단점을 살펴본다.

2.1. 가지치기방법

[2]에서는 다차원 인덱스의 일종인 R*-tree에 대하여 가지치기를 적용한 탐색 알고리듬이 소개되었는데, R*-tree에서 질의점이 주어졌을 때 질의 점과 최소경계사각형의 거리를 이용하여 최근접 질의를 처리하는 방법을 제안하였다.

이 방법은 주어진 질의 점과 하나의 최소경계사각형 사이의 최소거리를 MINDIST라고 정의하고, 노드를 방문할 때 MINDIST가 가장 작은 노드를 방문한다. 그러나 MINDIST는 그림 2와 같이,

질의 점에서 노드의 최소경계사각형과 가장 가까이 있어도, 노드 안에 실제 객체와의 최소 근접성은 보장하지 못한다. 따라서 실제 객체와의 최소 근접성을 보장할 수 있는 최소거리를 MINMAXDIST라고 정의한 다음, 후보노드를 방문할 때 MINDIST와 MINMAXDIST의 성질을 이용하여 방문할 후보 노드들을 그림2와 같이 가지치기하여 나감으로써 최근접 질의를 처리한다.

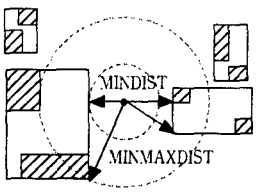


그림 2. 가지치기과정

2.2 문제점

이 가지치기방법은 다음과 같은 두가지 중요한 문제를 가지고 있다. 첫 번째 문제는 탐색영역을 최소화하지 못한다는 것이다. 가지치기 방법에서 성능을 높이는 가장 중요한 요인은 탐색되는 영역을 결정하는 반지름인 MINMAXDIST를 가능한 작게 만드는 것이다. 즉, 가능한 탐색되는 질의영역을 최소로 하여 노드의 방문을 줄이는 것이 매우 중요하다. 그러나, 가지치기 방법에 의하면, 그림3과 같이 노드의 최소경계사각형의 모양이나 위치에 따라, 최적의 반지름을 얻지 못하여, 불필요한 노드를 참조하고 결국 성능이 저하될 수 있다.

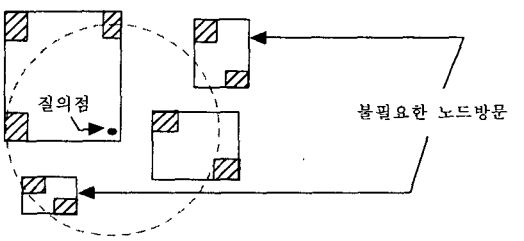


그림 3. 가지치기방법의 불필요한 노드방문

이 문제를 해결하기 위하여 [3]에서는 조사하는 탐색영역을 검색된 단말노드까지 검사된 공간객체들 중에서 최소거리로 결정하는 방법을 제안하였다. 그러나, 이 방법은 k-근접 객체를 찾기 위해 MINDIST가 가장 작은 노드를 선택할 때 항상 현재 노드의 자식 노드들만을 고려함으로써 불필요한 노드를 참조할 수 있다는 단점을 가지고 있다.

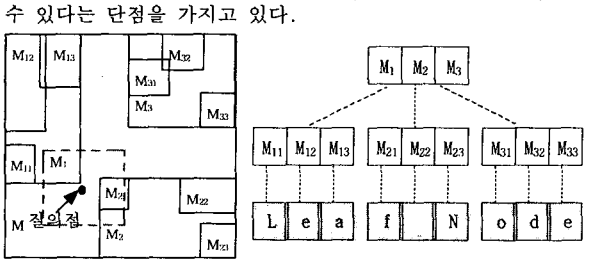


그림 4. 향상된 가지치기 방법

질의 점(q)에 대해 최근접 객체 탐색시, 기존의 가지치기방법을 이용하면 트리의 방문중, M_1, M_{11}, M_{12}, M_2, M_{21} 순서로 5번의 디스크 참조를 해야 한다. 뿐만 아니라 참조되어는 모든 노드와 실제거리를 계산해야 한다. 이는 불필요한 노드참조와 CPU계산을 포함하여 성능을 저하시킨다는 것을 의미한다.

두 번째로, 가지치기 방법은 최소의 탐색영역을 보장하지 못한다는 문제와 더불어, 탐색되는 영역이 가지치기과정 중에 동적으로 결정된다는 단점을 가지고 있다. 즉, 공간데이터베이스 시스템을 병렬화할 때, 동적으로 결정되는 탐색영역은, 부하의 분산을 결정하는 작업을 어렵게 한다.

위의 두 문제를 해결하기 위하여, 본 논문에서는 밀도를 이용하여 탐색영역을 결정하는 방법을 제안한다.

3. 밀도를 이용한 k-최근접 탐색 알고리듬

본 장에서는 밀도를 이용한 k-최근접 질의 알고리듬을 제시한다. 먼저 사용되어지는 기호에 대하여 정리를 하고, 밀도의 개념과 탐색 알고리듬을 살펴본다.

3.1. 밀도와 최근접 객체처리를 위한 탐색공간

가지치기방법에서 이용하는 탐색공간은 객체의 분포와 많은 관계가 있다. 객체가 조밀하게 분포된 지역에는 탐색공간은 작아지게 되며, 객체가 별로 없는 지역은 탐색공간이 넓어지게 된다. 본 논문에서는 이와 같은 특성을 고려하여, 탐색지역을 결정하는데 이용되는 반지름을 질의지역의 밀도를 이용하여 추정하는 방법을 제안한다.

3.2 알고리듬

본 논문에서 제안하는 최근접 객체의 질의처리방법은 크게 다음과 같이 네 단계의 작업으로 구성된다.

단계 1. 질의 지역에 대한 밀도의 추정
단계 2. 밀도를 이용한 탐색영역 반지름 추정
단계 3. 단계 2에서 구한 탐색영역으로 k-최근접 질의 처리
단계 4. 단계 3에서 k개의 객체가 찾아지지 않았을 경우, 탐색영역의 확대.

그러면, 위와 같은 방법으로 k-최근접 질의를 처리하는 과정을 자세히 알아보기로 한다. 아래 표는 알고리듬을 설명하면서 사용하게 될 기호와 함수에 대한 설명이다.

r	: 반지름
q	: 질의 점, $(q_1, q_2, \ldots q_d)$
R_d	: d차원에서 정다면체와 내접하는 다차원 구의 체적비
$D(q)$	: 질의영역 주위의 밀도
$A(q)$	: 초기선택률 추정을 위해 사용된 정다면체의 체적
$A(q,r)$	: 정다면체 $[q_i-r, q_i+r)^d$ 영역의 체적
$\hat{N}(q)$	: 초기선택률 추정에 의해 계산된 공간객체 수
$\hat{N}(q,r)$	: $[q_i-r, q_i+r)^d$의 영역에 대하여 선택률 추정된 공간 객체의 수
$N(q,r)$	: 질의 점(q)로 부터 거리 r이내에 있는 객체의 수

표 1. 기호와 함수 설명

[단계 1] 위의 방법을 적용하기 위해서는 우선 주어진 질의점의 밀도를 추정하여야 한다. 이 문제는 선택률 추정의 결과를 이용하여 계산될 수 있다. 선택률 추정방법에는 많은 방법이 제안되었으나, 본 논문에서는 자세한 언급은 생략한다. 질의영역 주위의 밀도는 다음과 같이 계산된다.

$$D(q) = \frac{\hat{N}(q)}{A(q)} \quad \cdots\cdots (1)$$

선택율 추정을 위해서 사용된 영역은 q를 중심으로 하는 적절한 정다면체이다.

[단계 2] 밀도가 위와 같이 계산되면 이를 이용하여 탐색영역의 반지름을 다음과 같이 계산할 수 있다.

$$r = \frac{1}{2}\sqrt[d]{\frac{k}{D(q)\,R_d}} \quad \cdots\cdots (2)$$

위의 식은 (1)를 이용하면 다음과 같이 된다.

$$r=\frac{1}{2}\sqrt[d]{\frac{kA(q)}{\hat{N}(q)\,R_d}} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (3)$$

그런데, 여기서 구한 r은 처음에 질의점(q)를 중심으로 임의로 정한 지역의 선택율을 이용한 것이다. q를 중심으로 균일분포가 아니면, 위의 (3)식은 수정되어야 한다. 즉, q를 중심으로 r만큼의 반지름을 가지는 정다면체를 이용하여 선택률 추정을 다시 한다. 이 과정은 다음과 같다.

단계 2-1. $r \leftarrow \frac{1}{2}\sqrt[d]{\frac{kA(q)}{\hat{N}(q)\,R_d}}$ 을 계산

단계 2-2. $[q-r, q+r)^d$의 영역에 대하여 선택률 $\hat{N}(q,r)$ 계산

단계 2-3. 만일 $|\hat{N}(q,r)R_d - k| > \varepsilon$ 이면,

$r \leftarrow r \cdot \sqrt[d]{\frac{k}{\hat{N}(q,r)R_d}}$로 하여 단계 2-2를 반복

위에서 $\varepsilon = k * 0.1$로 하여 계산하는 것이 효과적이라는 것을 실험적으로 발견할 수 있었다.

[단계 3] 위에서 구한 r로 주어지는 d차원의 구를 탐색영역으로 하여, 트리구조의 색인을 통하여 포함질의를 수행한다.

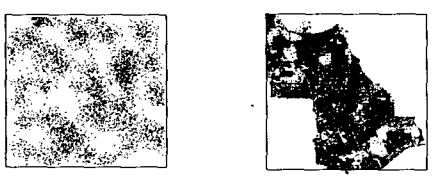
[단계 4] 만일 이 탐색결과로 k개 이상의 객체가 찾아지면 이 중에서 가까운 k개를 선택하면 최근접 질의처리의 결과가 된다. 그러나, 만일 탐색결과로 찾아진 객체의 수가 k개보다 작으면 단계 4의 과정을 거쳐야 한다. 그런데, 단계 4를 반복하게 되면, 질의처리성능이 매우 심각하게 저하된다. 따라서, 단계 4의 과정이 필요하게 되는 확률을 가장 최소로 하면서 동시에 반지름을 가능한 작게 만들기 위하여 δ 만큼 r를 증가시킨다.

$$r' = r + \delta \ , \ \delta = 2r(\sqrt[d]{\frac{k}{N(q,r)}} - 1)$$

4. 실험 및 성능 평가

본 장에서는 논문에서 제안한 알고리듬과 기존의 가지치기방법의 성능을 비교한다. 이 실험에서는 지금까지 트리구조 공간색인방법 중 가장 성능이 우수한 것 중에 하나로 알려진 R*-tree[1]를 사용하였다. 또한, 선택률 추정을 위하여 다차원 히스토그램을 이용하였다. 가지치기방법은 [3]에서 제안된 방법을 구현하였다.

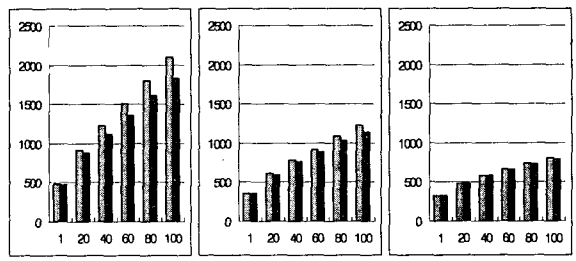
실험에 사용된 데이터는 2차원의 점 객체만을 대상으로 하였고, 10,000개의 점으로 이루어진 비균등한 합성 데이터와 36,548개의 점으로 이루어진 롱비치지역의 실제 데이터를 대상으로 실험하였다. R*-tree 구현 시 페이지 크기는 1K, 2K, 4K 바이트로 각각 실험하였다. k-최근접 질의에서 k는 1, 20, 40, 60, 80과 100으로 질의를 처리하였고, 질의는 100개의 임의로 생성된 2차원 점으로 수행하였다. 사용한 2차원 데이터는 다음과 같다.



(a) 합성된 비균등데이터 (b) 롱비치지역의 실제 데이터

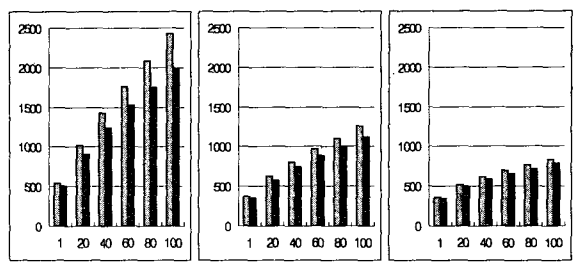
그림 5. 실험에 사용된 2차원 데이터

그림 6과 7은 위의 두 가지 데이터에 대한 실험결과를 나타낸다. 이 그림에서 세로축은 질의 100개에 대한 총 디스크 참조 횟수를, 가로축은 k의 값을 의미한다. 이 그림의 (a),(b)와 (c)는 각 1K, 2K, 4K의 페이지 크기를 말한다. 또한 그래프의 막대에서 왼쪽은 가지치기방법이고 오른쪽은 본 논문에서 제안한 밀도를 이용한 방법이다.



(a) 1K (b) 2K (c) 4K

그림 6. 합성된 비균등 데이터에 대한 성능비교



(a) 1K (b) 2K (c) 4K

그림 7. 롱비치 지역의 실제데이터에 대한 성능비교

위의 두 실험결과에서 보듯이 본 논문에서 제안한 방법이 기존의 가지치기방법에 비하여 약간의 성능이 향상되었다. 두 경우 모두 페이지의 크기가 작은 경우, 성능의 향상이 더욱 크다는 것을 알 수 있다.

5. 결론 및 향후 과제

본 논문에서는 많은 디스크 참조를 요구하는 k-최근접 탐색을 위한 새로운 기법을 제안하였다. 제안된 방법은 객체 분포에 대한 밀도를 이용하여 질의 조건을 만족하는 영역을 미리 예측함으로 불필요한 노드방문을 제거하였다.

실험결과를 통해, 제안된 방법이 최고 22%, 평균 7%의 디스크 참조횟수 감소의 성능향상을 보였다. 또한 제안된 방법은 질의처리 이전에 미리 탐색영역을 알 수 있어, 병렬화 등에 효과적으로 활용될 수 있다는 장점을 가지고 있다.

향후과제로는 현재는 디스크 참조횟수만을 성능평가의 기준으로 삼았지만, 질의 수행시 CPU처리시간도 비교 분석해야 할 것이다. 그리고 제안된 논문에서는 다차원히스토그램을 이용하여 구현하였지만, 보다 정확한 선택률 추정방법을 적용하면 성능이 더욱 향상되리라 예상된다. 또한 다차원으로 확장하여 실험하는 것과, 기존에 k-최근접을 위해 제안된 색인기법인 X-tree, SS-tree, SR-tree등을 이용해서도 성능 평가도 필요하다.

6. 참고 문헌

[1] N.Beckmann, H.Kriegel, R.Schneider, B.Seeger, " The R*-Tree: An Efficient and Robust Access Method for Points and Rectangles." , pp.322-331, Proc SIGMOD Conference, 1990,

[2] N.Roussopoulos, S.Kelley, F.Vincent," Nearest Neighbor Queries." , pp.71-79, Proc SIGMOD Conference, 1995.

[3] K.Cheung, A.Fu, "Enhanced Nearest Neighbour Search on the R-tree". pp16-21, SIGMOD Record 27(3), 1998