

# 이동 객체의 최근접 질의를 위한 유효 시간에 관한 연구

강구안\* 이상욱\*\* 김진덕\*

동의대학교 컴퓨터공학과\* 경상대학교 정보통신공학과\*\*

## A Study on Valid Time for Nearest Neighbor Query of Moving Objects

Ku-An Kang\* Sang-Wook Lee\*\* Jin-Doeg Kim\*

Donggeui University\* Kyungsang National University\*\*

E-mail : kangkuan@korea.com

### 요약

최근 눈부신 무선 통신 기술의 발전은 위치정보 시스템과 위치 기반 서비스(Location-Based Service)의 급속한 발전을 이루었다. 시공간 상에서 이동체에 대한 데이터베이스는 사용자에게 이동체의 현재위치 정보를 제공하는 것만큼 그 질의 결과의 유효시간 또한 매우 중요하다. 따라서 본 논문에서는 질의 점과 객체가 동시에 이동 중일 때 현재 질의 결과를 계산하는 방법과 그 질의 결과의 유효시간 및 유효시간 후의 질의 결과를 검색하는 방안을 제안한다. 이동 객체는 실시간으로 변화하기 때문에 현재 질의 결과가 조금만 시간이 지나도 잘못된 정보가 될 수 있고 미래의 결과를 반복연산에 의해 계산하기 어렵기 때문에 우리는 수학적식으로 유효 시간을 예측하고자 하는 것이다.

### ABSTRACT

The latest wireless communications technology bring about the rapid developments of Global Position System and Location-Based Service. It is very important for the moving object database to deal with database queries related to the trajectories of a moving objects and the valid time of the query results as well. In this paper, we propose how to get not only the current result of query but also the valid time and the result after the time when a query point and objects are moving at the same time. We would like to predict the valid time by formula because the current results will be incorrect due to the characteristic of the continuous movements of the moving objects and the future results can not be calculated by iterative computations.

### 키워드

최근접 질의, 유효시간, 이동체 데이터베이스

## I. 서론

최근의 무선 통신 기술, GPS(Global Position System)와 LBS(Location-Based System)의 눈부신 발전은 휴대폰과 PDA와 같은 휴대용 단말기의 사용이 일반화되면서 이를 응용한 서비스들이 보편화되고 있다. 즉 LBS와 GPS(Global Positioning System)의 측위기술을 이용하여 위치 추적과 위치 관련 정보를 실시간 제공할 수 있는 지능형 교통정보 시스템(ITS), 이동 중인 사용자가 현재 위치로부터 가장 가까운 택시를 찾는 서비스, 고속도로 주행 중에 일정한 시간 후에 나와 가장 가까운 주요소를

찾는 서비스와 같은 것이 실생활에 많이 이용되고 또한 많이 연구되고 있다[1, 2, 3, 6]. 따라서 시공간 상에서 이동체에 대한 데이터베이스는 연속적으로 이동하는 이동체의 위치 정보와 쿼리에 관련된 데이터베이스 질의를 효과적으로 처리 하여 사용자에게 이동체의 현재위치 정보를 제공하는 것은 아주 중요하며 시간이 지난 후의 이동체의 위치 정보 또한 아주 중요하다고 볼 수 있다. 왜냐하면 이동체의 특성상 현재 질의 결과가 조금만 시간이 지나도 위치의 변경으로 잘못된 정보가 될 수 있기 때문이다. 지금까지의 연구는 이동 객체 중 한쪽이 정적이고 다른 쪽이 동적인 것에 대한 연구와 현재 질의 결과를 찾기 위하여 거리 계산법을 사용하여 질의 결과를 제시하는 연구는 많이 이루어 졌다. 그러나 유효 시간을 이용한 연구는 미미한 수준으로 본 논문은 이동객체의 질의 결과를 시간이 지난 후에도 사용

\* 이 논문은 MOCIE 04' 지역특화기술개발사업(110015183)에 의하여 지원되었음

자에게 정확한 정보를 제공하기 위하여 유효시간 계산법으로 최근접객체(neighbor Nearest: NN)를 찾는 방법과 유효시간을 산출하기 위한 방법을 제안하고자 한다.

본 논문은 먼저 시공간 상에서의 이동 객체에 대한 데이터는 아주 방대하기 때문에 모든 이동 객체는 TPR-tree의 각각의 MBR이 관리한다고 본다. 그리고 시공간 상의 모든 객체를 검색하는 일반 알고리즘은 사용하지 않는다. 왜냐하면 이는 너무나 많은 검색 Overhead를 발생 시켜  $t_1$  시간 후의 정확한 최근접 질의 결과를 주기 어렵기 때문이다. 그래서 본 논문에서는 변형된 Heap 알고리즘[1, 4]을 사용하여 현재 질의 점에서 가장 가까운 Minmindist를 찾는 다음 이 MBR의 Minmindist를 찾는다. 이 값을 모든 MBR의 Minmindist와 비교하여 현재의 Minmindist보다 큰 Minmindist 값을 후보 MBR에서 제외한다. 이렇게 선택된 후보 MBR의 각각의 객체에 대하여 이동객체의 궤적정보를 사용하여 T 유효 시간을 구하는 새로운 방법을 제안하고자 한다.

논문의 구성은 다음과 같다. 2장에서는 본 논문에서 다루고자 하는 공간 데이터 베이스에서 이동객체의 최근접 질의에 관한 관련 연구에 대하여 기술하고, 3장에서는 연구 동기와 본 논문에서 제시하고자 하는 변형된 Heap 알고리즘으로 후보 MBR를 구하는 방법과 질의 객체로부터 이동객체까지의 유효 시간을 구하는 새로운 방법을 제안하였다. 4장에서는 결론을 맺는다.

### II. 관련연구

TPR-tree[1, 2, 3, 9]는 미래 위치를 예측하기 위해 R-Tree 기반에 속도 및 방향 정보를 추가한 것으로 이동객체의 위치 계산에 대표적으로 이용되는 방법이다. 삽입 삭제는 R-tree와 비슷하다. 이동 객체는 (i) 현재 시간에 확장된 MBR을 가지고 표현되거나, (ii) 속도 vector를 가지고 표현 된다. MBR은 이동객체의 최하위 속도와 최상위 속도를 가지고 MBR를 재구성함으로써 모든 이동 객체를 계속 MBR에 포함할 수 있다. 이동체의 최고 속도와 최저 속도 규칙을 이용하여 MBR의 크기를 변화시켜 시간이 지난 후에도 여전히 이동객체를 MBR에 포함하게 한다. TP질의는 시공간 상에서 연속적으로 이동하는 객체에 대한 질의 결과를 제공한다. TP 질의 결과는 기본적으로  $\langle R, T, C \rangle$ 를 가진다.

### III. 최근접 질의의 검색과 유효시간

#### 3.1 질의 점으로부터 후보 MBR 선택방법

본 논문에서는 객체가 모두 이동 중일 때 최근접 NN을 효율적으로 찾기 위하여 변형된 Heap 알고리즘을 응용하여 후보 MBR을 찾아 이 후보 객체만을 검색하여 검색의 Overhead를 줄이는 방법을 제

안한다. 또한 이 선택된 후보 MBR로부터 질의 객체로부터의 유효 시간 즉 이동객체가 질의 객체가 교차하는 시간인 유효시간을 함수식으로 구할 것이다. 먼저 시공간에서의 TPR 트리의 표준 출력은  $\langle R, T, C \rangle$ 이다. R은 현재 질의 결과를 나타내고, T는 현재 질의의 유효 시간 및 C의 결과가 어느 시점에 유효하고 어느 시점까지 유효 한지를 나타내는 유효 시간이고 C는 R의 다음 NN의 결과를 나타낸다[1, 2, 3].

본 논문에서는 검색의 Overhead를 줄이기 위하여 기존의 Heap 알고리즘을 변형하여 후보객체가 되는 MBR를 선정 한다. 기존 연구에서는 질의 점 q로부터 모든 Minmindist의 값을 구하여 최소값의 Minmindist를 후보 MBR로 선정 했다. 그러나 이것은 정확한 결과를 얻지 못할 수도 있다. 왜냐하면 현재 질의 점 q에서 다음 시간에 현재 질의 점 q에서 아주 멀지만 객체 중에 아주 빠른 속도로 움직이는 MBR도 후보 객체로 검색해야 하고 또한 MBR이 중첩이 되는 경우는 비록 현재의 Minmindist가 멀지만 질의 점 q에 가까운 NN이 존재할 수 있기 때문이다[1, 3, 6]. 그림 1은 MBR a, b, c 그리고 d의 Minmindist를 계산하여 현재 질의점 q에 가장 가까운 MBR b의 Minmindist를 구하여 모든 MBR의 Minmindist와 비교하여 후보 MBR를 선정하는 방법을 제안하고 그 후보 검색 MBR의 보기이다.

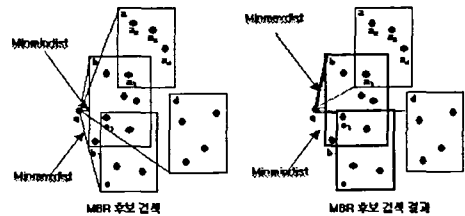


그림 1. q의 Minmindist와 Minmaxdist

그림1에서 기존의 연구는 질의 지점 q로부터 가장 가까운 MBR의 Minmindist의 거리를 계산하여 질의 점 q로부터 가장 가까운 Minmindist를 후보객체로 선정하여 MBR b안의 모든 후보 객체의 거리를 계산하여 b이 현재 질의 결과라는 잘못된 결과를 출력하는 오류를 범할 수 있었다. 따라서 그림 1에서의 NN의 출력 결과는 b이 아닌 c1이라는 것을 그림으로 보여 주는 것이다. 따라서 그림 1은 후보 MBR의 중첩 및 속도를 고려하지 않으면 정확한 결과를 보여 주지 못한다는 것이 증명되는 것이다. 따라서 본 논문에서는 보다 정확한 현재 질의 결과를 찾기 위하여 이러한 문제를 변형된 Heap 알고리즘을 사용하여 후보 객체를 선정하고자 한다.

1. D를 초기화 한다.
2.  $D = \text{Find}(q, \text{Minmindist})$  // 현재 질의 q로부터 가장 가까운 MBR를 Minmindist값을 찾아 이를 D에 저장한다.
3.  $D = \text{Find}(\text{Minmindist}, \text{Maxmaxdist})$  // 현재 질

의 점  $q$ 의 가장 가까운 MBR의 Minmaxdist를  $D$ 에 저장한다.

4. Loop 모든 MBR의 개수만큼
  - If  $D \geq$  Each MBR Minmaxdist then
  - $D =$  MBR Minmaxdist
  - End IF
  - // 최소값의 Minmaxdist를 구한다
5. Loop 모든 MBR의 개수 만큼
  - If  $D \geq$  Each MBR Minmindist then
  - 후보검색 MBR로 등록
  - End IF
  - EndLoop
  - //  $q$ 의 검색 대상이 되는 MBR의 후보 객체를 찾아 저장한다.

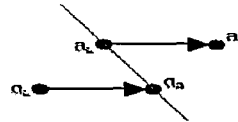


그림 2. 이동객체간의 교차점의 정의

유효 시간을 구하기 위해서는 우리는 질의 객체와 이동 객체가 교차하는 시간의 값을 구한다. 예를 들면 이동 객체와 질의 객체가 모두 오른쪽으로 이동 중 일때 이동객체의 궤적  $t_1$ 이 질의 객체의  $t$ 와 교차하는 시간을 유효시간으로 정의할 수 있다. 그림 3은 속도를 가진 질의 객체의 시간 후의 유효 시간을 구하는 방법을 제안하였다.

### 3.2 현재 최근접 질의 결과 추출법

3.1절에서 제안된 알고리즘을 사용하여 선택된 후보 MBR은 R-tree를 사용하여 현재 이동 객체들의 정보를 저장한다. 그리고 Heap 알고리즘을 사용하여 거리를 최소값 형식으로 계산 결과를 Heap에 저장한다. 이 저장된 Heap의 값을 차계대로 가지고 오면 현재 질의 결과를 얻을 수 있다. 그러나 이는 현재 질의의 결과는 이동 객체의 속도를 고려하지 않아서 속도를 가진 이동 객체의 연속적이 질의 결과에는 정확한 결과가 될 수 없다. 이 결과 값에 이동 객체의 속도를 고려한 함수식을 적용 함으로서 이동객체의 정확한 연속 최근접 NN를 찾을 수 있다.

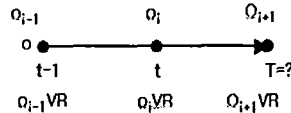


그림 3. 궤적 정도를 이용한 유효시간

이동객체는 실시간으로 이동하기 때문에 미래  $T$ 에 도착하는 시간을 예측해야 한다. 왜냐하면 현재 질의 결과는 모든 객체를 고정객체의 값으로 계산하여 현재 결과를 출력하지만 실제로 이는 질의를 입력하여 출력이 되는 순간은 정확한 정보이지만 시간이 약간 변하면 정확한 정보가 될 수 없다. 따라서 본 논문에서는 미래 시간 즉 언제 객체가  $T=?$ 에 도달하는 지가 아주 중요하다. 왜냐하면 현재 질의 결과가 얼마간 유효 하며 시간이 지나면서 다음 질의 결과를 함수식으로 계산 하여 구할 수 있다 때문이다. 그림 4에서 미래  $T=?$ 에 도착하는 시간을 계산하는 식은 이동객체의 현재 질의 시간  $t$ 의 궤적정보  $t_1$ 를 이용하여 구할 수 있다.

### Function(o, Minmaxdist)

1. R-tree의 루트로부터 시작하여,  $D$ 를  $\infty$ 로 설정하고 힙트를 초기화 한다
2. Loop till end of R-tree node
3. If leaf node then
  - 4.  $D =$  거리 계산
  - 5. Heap =  $D$  알고리즘에 저장
6. End If
7. End Loop
8. return  $D$

현재 질의의  $q$ 로부터 R-tree에 저장된 값을 중간 노드가 아니면 거리를 계산하여 Heap에 저장한다. 만약 모든 노드까지 계산이 끝나면 Heap안의 내용을 차계대로 가지고 오면 NN의 결과를 가지고 올 수 있다.

### 3.3 이동객체의 유효시간

이동 객체간의 교차점의 정의는 그림2에서와 같이 이동객체  $a_L$ 이 질의점  $q_R$ 을 교차하는 점으로 정의 한다. 따라서 객체  $a$ 가 질의 점  $q$ 를 교차하는 유효 시간은  $a$ 이  $q_R$ 를 교차하는 시간이 다.

$$T = t_{+1} = \frac{q_i - q_{i-1}}{q_i VR - q_{i-1} VR}$$

위의 식은 질의 점 객체  $q_1$  이 현재 시간의  $q$  지점에 도착하는 시간을 구하는 함수식으로 표현한 것이다. 즉 현재 질의 지점  $q$ 의 이동거리를  $q$ 가 가지는 속도 차로 나누어  $T=q_{i+1}$ 의 시간을 구하는 함수식이다. 이 방법을 각각의 이동 객체에 적용하여 현재 질의로부터 각각의 이동객체까지의 유효 시간을 구할 수 있다.

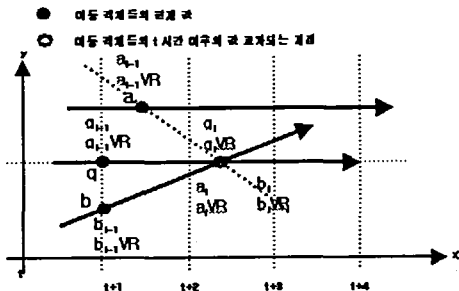


그림 4. 질의점 q와 객체 a와 b의 유효 시간

질의 객체 q로부터 각각의 이동객체 a와 b의 유효 시간은 다음과 같은 식으로 구할 수 있다. 이동객체 a와 q는 평행하기 때문에 결코 만날 수 없다. 그러나 유효시간은  $a_{t-1}$  이  $q_t$ 와 교차하는 시간을 구하는 함수식이다.

$$T_{a,q} = \frac{q \cdot q}{q_{a,q} \cdot q_{a,q} \cdot \frac{q_{a,q} - q_{a,q}}{q_{a,q}VR - q_{a,q}VQ}}$$

이동 객체 b는  $b_{t-1}$  시간이  $q_t$ 와 교차하는 시간을 구하는 유효 시간 함수식은 다음과 같이 구할 수 있다.

$$T_{b,q} = \frac{q \cdot q}{q_{b,q} \cdot q_{b,q} \cdot \frac{q_{b,q} - q_{b,q}}{q_{b,q}VR - q_{b,q}VQ}}$$

위에서 표현된 유효시간 함수를 앞에서 현재 질의 결과를 찾는 함수식에 적용하면 연속적인 질의 결과를 얻을 수 있다.

TFunction(o, Minmaxdist) // 유효 시간 적용 알고리즘

1. R-tree의 루트로부터 시작하여, D를 ∞로 설정하고 히프를 초기화한다
2. Loop till end of R-tree node
3. If leaf node then
4. D = 거리 계산 \* T (유효시간)
5. Heap 알고리즘에 저장
6. End If
7. End Loop
8. return D

위의 알고리즘을 수행하면 Heap에는 유효시간 후의 이동객체들이 정렬되어 저장된다. 이것을 k개 만큼 출력하면 현재 질의로부터 가까운 k개의 질의 결과를 찾을 수 있다.

#### IV. 결론

이동 객체는 시간에 따라 그 위치가 시시각각 변화한다. 기존 연구에서는 이동객체가 한쪽이

정적인 질의는 많이 연구 되어 왔다. 그러나 모든 질의 객체가 이동 중인 경우의 연구는 많이 없는 편이다. 따라서 본 논문에서는 기존에 연구된 Heap 알고리즘으로 질의 지점 q에 가장 가까운 MBR의 Minmaxdist를 구한다음 각각의 MBR의 Minmindist와 비교하여 Minmaxdist보다 큰 Minmindist는 후보 객체에서 제외하므로 비교 검색의 Overhead를 크게 줄였다. 또한 후보 객체로부터 질의 결과를 Heap알고리즘에 저장하여 현재질의를 구하였고, 유효 시간계산은 이동 객체의 궤적정도를 이용하여 유효시간을 유도할 수 있는 수식을 만들었다. 이 연구는 객체가 이동중인 서비스에 적합한 이론이다. 예를 들면 지역형 맞춤 광고 즉 이동용 차량이 어느 지역을 지나면서 현재 이동 객체와 가까운 이동객체의 멤버들에게만 광고를 전달 수 있는 응용분야와 사용자가 움직이면서 택시를 호출하여 현재 나와 가장 가까운 택시에게 사용자의 현재 위치를 알려 택시가 바로 사용자에게 갈수 있게 하는 분야 등 많은 부분에 응용이 가능하다.

이번 논문에서는 이동 객체의 NN을 구하는 방법만 제시였다. 다음 연구에는 실제 이론만 제시하는 것이 아니라 실제로 데이터를 입력하여 성능을 분석하고, 실생활에 사용이 가능하게 실제 구현을 하고 상품으로까지 만드는 것에 도전하는 것이 최종 목표이다.

#### 참고문헌

- [1] Y. Tao and D. Papadias, *Time-Parameterized Queries in Spatio-Temporal Databases*, ACM SIGMOD, 2002
- [2] P.K. Agarwal, L. Arge, and J. Erickson, *Indexing Moving Points*, PODS, 2000
- [3] 권영철, 배진욱, 이석호, "이동객체 데이터베이스에서 TP 최근점 쌍 질의의 처리" 한국정보과학회, 2003
- [4] 전봉기, 임덕성, 홍봉희 "이동체 데이터 베이스를 위한 색인 기법" 한국정보과학회, 2001
- [4] S. Acharya, V. Poosala, and S. Ramaswamy, *Selectivity Estimation in Spatial Databases*, ACM SIGMOD, 1999
- [5] N. Beckmann, H.P. Kriegel, R. Schneider, and B. Seeger, *The R\*-Tree: An Efficient and Robust Access Method for Points and Rectangle*, ACM SIGMOD, 1999
- [6] K. Porkaew, I. Lazaridis, and S. Mehrotra, *Querying Mobile Objects in Spatio-Temporal Databases*, SSTD, 2001
- [7] S. Saltenis and C.S. Jensen, *Indexing of Moving Objects for Location-Based Services*, ICDE, 2002
- [8] S. Saltenis, C.S. Jensen, S.T. Leutenegger, and M.A. Lopez, *Indexing the Positions of Continuously Moving Objects*, ACM SIGMOD, 2000
- [9] Y. Theodoridis, E. Stefanakis, and T.K. Sellis, *Efficient Cost Models for Spatial Queries Using R-Trees*, TKDE, 2000