

# Time Slice에 따라 인덱스를 분할하는 이동경로 인덱싱 기법

강현민 김기홍 차상균  
서울대학교 전기공학부

## Trajectory Indexing Technique with Time-sliced Index Partitioning

Hyun M. Kang Kihong Kim Sang K. Cha  
School of Electrical Engineering, Seoul National University

### 요약

시공간 상에서 움직이는 물체의 이동경로는 일반적으로 MBR이 매우 크므로, R-tree 계열의 다차원 인덱싱 기법을 적용할 때 질의시 겹침 영역이 많이 발생하는 문제점이 있다. 본 논문에서는 이러한 문제점을 해결하기 위해 time slice에 따라 이동경로의 MBR을 나누어 별도의 인덱스에 저장하는 기법을 제안한다. 이 기법을 적용하면 객체 MBR의 평균 크기가 줄어들어 인덱스 페이지 및 데이터 페이지 접근횟수를 감소시킬 수 있을 뿐만 아니라, 시간이 지나 질의가 매우 드물게 수행되는 데이터에 대한 엔트리를 삭제하여 검색 범위를 줄이는 것이 용이하다. 또한 본 논문에서는 질의 성능을 최적화하기 위해서 time slice를 어떻게 설계해야 하는지에 대해서도 논의한다.

## 1 서론

최근 급속하게 발전하고 있는 이동 통신 및 위성 기술은 GPS(Global Positioning System) 등을 이용하여 움직이는 물체의 위치를 확인하는 것을 가능하게 하였다. 이에 따라 움직이는 물체의 위치 및 이동경로(trajecory)를 관리하는 시스템에 대한 필요가 증대되고 있다. 특히 움직이는 물체의 이동경로는 시간에 따라 지속적으로 그 양이 증가해 방대한 양이 되므로 이를 짧은 시간에 효율적으로 검색하기 위한 인덱스 구조가 매우 중요하다.

그러나 움직이는 물체의 이동경로는 긴 시간과 넓은 공간에 걸쳐 있는 경우가 많아, 이를 하나의 MBR로 근사하여 R-tree 계열의 공간 인덱스를 적용하면, 질의시 겹침 영역이 지나치게 많이 발생하여 결과적으로 인덱스의 검색성능이 저하된다는 단점이 있다. MBR의 평균 크기가 증가할수록 질의 영역이 MBR과 겹칠 확률은 지속적으로 증가하게 되어 인덱스 페이지의 접근횟수가 증가하게 된다. 뿐만 아니라, LineString으로 이루어진 객체의 MBR은 크기가 증가함에 따라 빈 공간이 많아져 MBR 근사가 부정확하게 된다. 따라서 실제 객체는 질의영역과 겹치지 않으나 객체의 MBR이 질의 영역과 겹쳐 불필요하게 데이터 페이지를 접근해야 하는 false hit이 많아진다는 단점도 존재한다[1].

이러한 이동경로 인덱싱에 있어서의 문제점을 해결하기 위해 본 논문에서는 time slice에 따라 MBR을 분할하여 별도의 인덱스에 저장하는 기법을 제안한다. 이 기법을 적용하면 객체 MBR의 평균 크기가 줄어들어 인덱스 페이지 및 데이터 페이지 접근횟수를 감소시킬 수 있을 뿐만 아니라, 시간이 지나 질의가 매우 드물게 수행되는 데이터에 대한 엔트리를 삭제하여 검색 범위를 줄이는 것이 용이하게 된다.

본 논문의 2절에서는 본 논문에서 다루는 이동경로 인덱싱 문제를 정의하고 관련된 최근의 연구를 살펴본다. 그리고 3절에서는 본 논문에서 제안하는 이동경로 인덱싱 기법에 대해 설명하고, 이에 대한 구체적인 알고리즘을 기술한다. 4절에서는 효율

적인 검색성능을 얻기 위해서 어떠한 기준으로 time slice를 설계해야 하는지에 대해 분석적으로 기술하고 5절에서는 결론과 향후 계획에 대해 기술한다.

## 2 문제 정의 및 관련 연구

### 2.1 이동경로 인덱싱 문제 정의

본 논문의 대상이 되는 물체의 이동경로는 주로 GPS를 이용하여 확인된 물체의 위치 및 시간 정보를 이용하여 구성된 데이터라고 가정한다. 구체적으로 데이터는 다음과 같은 특성을 가진다.

- 물체는 2차원 평면 상의 점으로 표현되며 임의의 방향과 속도로 이동할 수 있다.
- 물체는 불연속적인 시각에 자신의 ID, 현재 위치 및 시각을 report한다.
- 물체가 report하는 위치 및 시각의 오차는 무시할 수 있다.
- 물체의 이동경로는 3차원의 시공간 상에서 불연속적인 점들을 잇는 선분이다.

이러한 특성을 갖는 물체의 이동경로를 이용하여 수행할 수 있는 질의 및 연산은 영역 질의, k-nearest 질의, 시공간 조인 연산 등 다양한 종류가 있다. 이 중 본 논문에서는 우선적으로 영역 질의를 효과적으로 수행할 수 있는 인덱스 구조 및 알고리즘에 초점을 맞추어 기술하였다.

### 2.2 관련 연구

MOD(Moving Object Database)에 대한 연구는 1990년대 후반부터 MOST 및 FTL[2], CHOROCHRONOS[3] 등의 연구 과제들을 통해 활발해지기 시작했다. 이동경로 인덱싱은 MOD에 대한 여러 연구 중 핵심적인 한 부분이나, 아직 많은 연구가 진행되지 않았다. 우리가 알고있는 바로는 RT-tree[4], HR-tree[5], Overlapping Linear Quadtree[6] 정도 뿐이다.

그러나 이러한 기존의 연구들은 전통적인 시간 DB에 사용되던 일반적인 관점을 고수하여 움직이는 객체는 다음 report가 있을 때까지 이전 위치에 머물러 있다가 report가 일어나는 순간 위치의 비약이 일어난다고 가정하는데, 이는 일반적인 이동경로의 의미와 다르다. 본 연구에서는 움직이는 객체의 이동경로를 report된 두 점 사이의 선분으로 정의하여 일반적인 의미와 가까운 이동경로 표현 및 이에 대한 인덱싱을 가능하게 하여 기존 연구와 차별성을 갖는다. 관련된 다른 연구로는 움직이는 물체의 위치와 속도정보를 가지고 현재 혹은 미래의 위치를 예측하여 이루어지는 질의를 위한 인덱스가 [7]등에서 제안되었지만, 이동경로 인덱싱 문제와는 그 대상이 다르다.

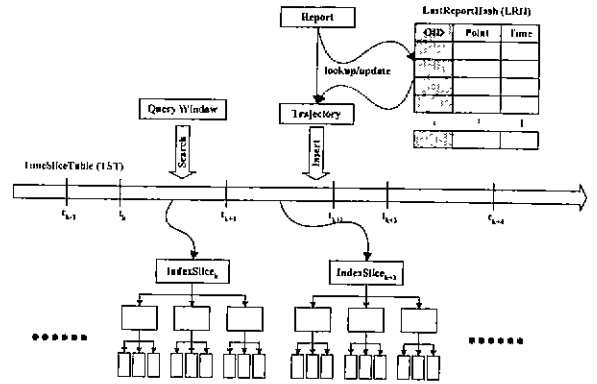


그림 1. 이동경로 인덱스의 전체 구조

### 3 Time-sliced Index를 이용한 이동경로 인덱싱

#### 3.1 기본 아이디어

공간상에서 움직이는 물체의 이동경로를 인덱싱 하기 위해 본 논문에서 제안한 기법의 기본 아이디어는 다음과 같이 두 가지로 정리할 수 있다.

##### 1. Time slice에 따른 별도의 인덱스 구축

물체의 이동경로를 하나의 3차원 인덱스에 저장하는 것은 앞서 언급한 바와 같이 겹침영역이 많이 발생하여 인덱스 및 데이터 페이지 접근횟수가 늘어난다는 단점과, 질의가 드물게 수행되는 엔트리를 삭제하기 어렵다는 단점이 있다. 이러한 문제점을 개선하기 위해 본 논문에서는 시간축을 적절히 분할하여 time slice를 만들고 각 time slice마다 별도의 인덱스를 구축하는 방법을 사용하였다.

##### 2. 각각의 IndexSlice를 2D 인덱스로 구축

가정에 따라 물체의 한 이동경로는 여러 개의 time slice를 거치는 것이 일반적이므로 한 time slice내에 들어 있는 MBR의 시간축을 살펴보면 대부분의 경우에 해당 time slice 전체를 포함하게 된다. 그 결과로 한 time slice내에서의 시간축은 MBR들간을 구분하는 주요 요소가 되지 못하게 된다. 따라서 본 논문에서는 각각의 IndexSlice를 구축할 때에 시간축을 고려하지 않은 2D 인덱스로 구축하여 저장 효율을 높임으로써 검색 성능을 향상시킨다.

#### 3.2 인덱스 구조

그림 1은 제안된 이동경로 인덱스의 구조를 보여준다. 긴 시간에 걸친 물체의 이동경로의 MBR은 분할되어 여러개의 IndexSlice(IS)에 분산 저장된다. 이렇게 time slice에 따라 분산되어 있는 IS들을 상위 레벨에서 관리하기 위한 자료구조가 필요한데, 이것이 TimeSliceTable(TST)이다. TST는 각 IS의 ID 및 시작 시간의 배열로 이루어져 서로 다른 시간 간격을 가진 time slice들을 수용할 수 있고, 모든 삽입, 삭제, 질의 수행은 이 TST를 통해 이루어지게 된다. LastReportHash(LRH)는 객체 삽입에 필요한 자료 구조이다. GPS를 통해 report된 위치 및 시간정보를 이용해서 이동경로를 만들기 위해서는 가장 최근에 report된 위치 및 시간정보를 알아야 할 필요가 있다. 이를 위해 LRH는 각 객체가 가장 최근에 report한 위치 및 시간정보를 객체 ID를 key로 갖는 hash로 저장하여 빠르게 이동경로를 만들 수 있게 해 준다.

#### 3.3 세부 알고리즘

Time slice에 따라 분할된 인덱스인 IS에는 R-tree 기반의 기존의 공간 인덱스를 그대로 사용할 수 있다. 따라서 여기에서는 IS에 객체를 삽입하거나 검색하는 레벨의 알고리즘들은 다루지 않고, 그 상위 레벨에서의 삽입 및 검색 알고리즘만을 다루도록 한다. 단, 아래의 모든 알고리즘에서  $r$ 은 report를,  $lr$ 은 last\_report를 의미한다.

##### 3.3.1 삽입 알고리즘

1. (OID,  $P_r$ ,  $t_r$ )정보가 포함된 report정보를 받는다.
2. OID로 LRH를 검색하여  $P_r$ ,  $t_r$ 에 결과를 넣는다.
3. [ $t_r$ ,  $t_r$ ]로 TST를 검색하여 관련된 IS를 찾아 ISQ(index queue)에, time slice가 일어나는 시각들을 찾아 TQ(time queue)에 넣는다.
4. TQ가 비어있으면 ISQ의 첫 인덱스에  $MBR(P_r, P_r)$ 을 키로 하는 객체를 삽입 또는 갱신하고 알고리즘을 종료한다.
5. 그렇지 않으면 TQ에서 시각을 pop하여  $t_{slice}$ 에 넣고, 이에 알맞은  $P_r, P_r$ 의 내분점을  $P_{slice}$ 에 넣는다.
6. ISQ에서 인덱스를 pop하여  $MBR(P_r, P_{slice})$ 를 키로 하는 객체를 삽입 또는 갱신한다.
7.  $P_r = P_{slice}$ 로 하고 4의 과정을 반복한다.

##### 3.3.2 검색 알고리즘

1. 질의 윈도우의 시작 및 끝 시간을 구해 각각  $t_{start}$ ,  $t_{end}$ 라 한다.
2. [ $t_{start}$ ,  $t_{end}$ ]로 TST를 검색하여 관련된 IS를 찾는다.
3. 관련된 모든 IS에 질의 윈도우를 이용해 질의를 수행한다.
4. 질의 결과를 종합하여 되돌려준다.

### 4 검색성능 최적화를 위한 time slice의 설계

본 논문의 핵심 아이디어는 물체의 이동경로의 MBR이 지나치게 크면 겹침 영역이 많이 발생하므로 이를 줄이기 위해 MBR을 분할한다는 것이다. 그러나 지나치게 MBR을 많이 분할하게 되면 겹침 영역은 그다지 감소하지 않는 반면에, 인덱스 엔트리의 수가 증가하게 되어 오히려 검색성능이 저하될 수 있다. 따라서, 인덱스의 검색성능을 최적화하기 위해서는 이러한 두 가지 측면을 고려하여 MBR이 적절한 크기로 분할되도록 time slice를 설

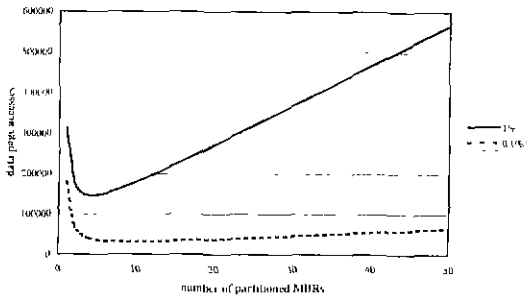


그림 2 분할된 MBR의 개수에 따른 데이터 접근횟수

계하는 것이 중요하다. 이러한 time slice 간격  $\Delta T$ 는 다음과 같이 구할 수 있다.

먼저, 단위 부피의 3차원 공간 상에서 MBR의 평균 부피를  $\bar{a}$ 라고 하고, 질의 영역의 크기를  $s$ , 객체의 총 개수를  $N$ 이라 하자. 이때 MBR이  $k$ 개로 분할되면, MBR을 이루는 객체가 선분이라고 가정할 때 MBR의 평균 부피는  $\bar{a}/k^3$ 이 되므로 한 MBR이 질의 영역에 들어갈 확률  $P(k, s)$ 는 [8]에 의해 다음과 같이 정해진다.

$$P(k, s) = \left( \frac{\sqrt[3]{\bar{a}}}{k} + \sqrt[3]{s} \right)^3 \quad (1)$$

분할 후 객체의 총개수는  $Nk$ 가 되므로 질의 영역과 MBR이 겹치는 객체의 총개수  $A(k, s)$ 는

$$A(k, s) = NkP(k, s) = Nk \left( \frac{\sqrt[3]{\bar{a}}}{k} + \sqrt[3]{s} \right)^3 \quad (2)$$

가 된다.

그림 2은 단위 부피의 3차원 공간 상에서 평균 부피가 0.1인 MBR 100단계를 각각  $k$ 개로 분할했을 때 1% 및 0.1%영역질에 대하여 접근하는 데이터의 개수를 나타낸 그래프이다. 그림에서와 같이 MBR을 지나치게 적게 분할하거나 많이 분할하게 되면 검색성능이 저하되는 것을 볼 수 있다. 최적화된 검색성능을 얻기 위한  $k_{optimal}$ 은  $A(k, s)$ 가 최소가 되는  $k$ 이므로 다음과 같이  $A(k, s)$ 를 미분하여 구할 수 있다.

$$\frac{dA(k, s)}{dk} = N \left( \frac{\sqrt[3]{\bar{a}}}{k} + \sqrt[3]{s} \right)^2 \left( \sqrt[3]{s} - \frac{3\sqrt[3]{\bar{a}}}{k} \right) \quad (3)$$

이 미분식에서  $A(k, s)$ 가 극소가 되는 점이 곧 최소가 되는  $k_{optimal}$ 이고 그 값은 다음과 같다.

$$k_{optimal} = \frac{3 - \sqrt[3]{\bar{a}}}{\sqrt[3]{s}} \quad (4)$$

따라서, 효과적인 인덱스의 검색성능을 위해서는 MBR의 평균 크기와 질의 영역의 평균적인 크기에 따라  $k_{optimal}$ 을 구해 이에 알맞게 time slice 간격을 정하는 것이 효과적일 것으로 기대된다. 단, 더 정밀한 결과를 위해서는 시간축의 특수성을 고려하여 위의 식을 재구성할 필요가 있다.

## 5 결론 및 향후 계획

이동경로 인덱싱 문제는 무선통신 및 위성 기술의 발전에 따라 중요한 문제로 부각되고 있다. 그러나 이동경로 인덱싱과 관련된 몇 안되는 연구들은 일반적인 이동경로의 의미를 제대로 표

현하지 못한다는 문제가 있고, R-tree 계열의 다차원 인덱싱 기법을 적용하면 객체 MBR이 너무 커 질의 시점 영역이 많이 발생하여 인덱스 페이지 및 데이터 페이지 접근횟수가 크게 증가하게 된다. 본 논문은 이동경로 객체 MBR의 겹침 영역을 줄이기 위해 MBR을 time slice로 분할하고 이들을 별도의 인덱스에 나누어 저장하는 방법을 제안한다. 이 기법을 사용할 때, MBR의 평균 크기와 평균 질의영역 크기를 고려하여 time slice간격을 적절히 설정해 준다면 최적화된 검색성능을 얻을 수 있다.

향후에는 먼저 이 기법을 기반으로 한 k-nearest 질의, 시공간 조인 연산 등의 알고리즘을 설계 및 구현하고 이를 영역질의 알고리즘과 더불어 [9]의 데이터 등을 이용해 실험할 예정이다. 또한 time slice의 간격을 더 길게 설정하는 대신 한 time slice내에서 객체 MBR을 분할해서 저장할 수 있도록 하여 서로 속도가 다른 객체들을 효과적으로 다룰 수 있도록 하는 방향으로 연구결과를 발전시켜 나갈 예정이다. 또한 실제 응용에 사용될 때의 인덱스 갱신 비용의 부하를 줄이기 위한 휴리스틱을 적용할 계획도 가지고 있다.

## 참고문헌

- [1] 강원민, 차상균, "RR-tree : 한 방향으로 긴 공간 객체에 대한 효율적 색인 기법," in 제 26회 정보과학회 봄 학술발표 논문집, pp. 113-115, 1999.
- [2] P. Sistla, O. Wolfson, S. Chaberlain, and S. Dao, "Modeling and Querying Moving Objects," in 13th IEEE International Conference on Data Engineering, pp. 422-432, 1997.
- [3] T. Sellis et al, "http://www.dbnet.ece.ntua.gr/ choros/." The CHOROCRONOS Project.
- [4] X. Xu, J. Han, and W. Lu, "RT-tree: An Improved R-tree Index Structure for Spatiotemporal Databases," in 4th International Symposium on Spatial Data Handling, pp. 1040-1049, 1990.
- [5] Y. Theodoridis, T. Sellis, A. Papadopoulos, and Y. Manopoulos, "Specifications for Efficient Indexing in Spatiotemporal Databases," in 10th International Conference on Scientific and Statistical Database Management, pp. 123-132, 1998.
- [6] T. Tzouramanis, M. Vassilakopoulos, and Y. Manopoulos, "Overlapping Linear Quadrees : A Spatio-temporal Access Method," in 6th ACM International Workshop on Geographical Information Systems, pp. 1-7, 1998.
- [7] G. Kollios, D. Gunopulos, and V. Tsotras, "On Indexing Mobile Objects," in 18th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, pp. 261-272, 1999.
- [8] C. Faloutsos and I. Karmel, "Beyond Uniformity and Independence: Analysis of R-trees using the Concept of Fractal Dimension," in 13th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, pp. 4-13, 1994.
- [9] Y. Theodoridis, J. Silva, and M. Nascimento, "On the Generation of Spatiotemporal Datasets," in 6th International Symposium on Large Spatial Databases, pp. 147-164, 1999.