

이동객체 데이터베이스에서 TP 최근접 쌍 질의의 처리*

권영철⁰ 배진욱 이석호

서울대학교 컴퓨터공학과

{quanyz⁰,oblody@db.snu.ac.kr}, shlee@cse.snu.ac.kr

Time Parameterized Closest Pair Queries in Moving Object Database

Yongzhe Quan⁰, Jinuk Bae, Sukho Lee

Dept. of Computer Science and Engineering Seoul National University

요 약

최근 들어 위치정보 시스템(GIS)의 발전으로 움직이는 물체의 위치는 쉽게 알 수 있게 되어서 이를 기반으로 미래의 최근접 쌍을 찾는 질의가 중요하게 되었다. 하지만 이동객체는 계속해서 움직이므로 현재의 질의 결과는 시간이 흐름에 따라 유효하지 않게 된다. 본 논문에서는 사용자에게 현재의 가장 인접한 쌍과 이 인접한 쌍이 유효한 시간, 그리고 그 유효한 시간 후에 바뀐 결과를 알려주는 질의를 효율적으로 처리할 수 있는 알고리즘을 제안한다.

1. 서 론

최근에 무선 통신이나 GPS와 같은 기술들의 비약적인 발전으로 인해, 시공간 데이터베이스가 많이 연구되고 있다. 시공간 데이터베이스에서의 질의는 주로 두 가지 형태로 분류할 수 있다. 하나는 과거 위치를 위한 질의이고, 다른 하나는 이동 객체의 미래의 위치를 위한 질의이다. 그 중에서 이동객체의 미래의 위치나 정보에 관한 질의들이 많이 다루어지고 있다. 본 논문에서는 특히 이동객체의 미래의 최근접 쌍(closest pair)을 찾는 질의를 주로 다루고자 한다.

TP(Time-Parameterized) 최근접 쌍 질의의 예로는 다음과 같다.[1] "시간에 따라 변화하는 가장 가까운 거리에 있는 택시는 무엇인가?"와 같은 것이 있다. 일반적인 방법으로 이런 질의를 수행하려면 질의 결과가 바뀌는 시각을 찾아낼 때까지 매 시각마다 인접한 쌍을 찾는 질의를 던져야 하므로 이는 매우 비효율적이다.

최근접 쌍 질의를 다루면서 현재의 질의 결과도 중요하겠지만 현재 질의 결과의 유효시간도 중요하다. 본 논문에서는 이동객체의 가장 인접한 쌍을 찾는 질의를 처리할 때 사용자들에게 현재의 질의 결과를 제공해주는 동시에 현재 질의의 유효시간과 유효시간 후에 바뀌는 질의 결과도 함께 반환해줄 수 있는 알고리즘을 제시하고자 한다.

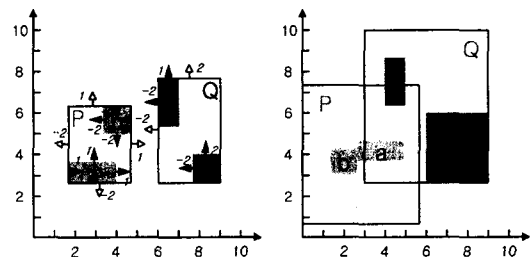
논문의 순서는 아래와 같다. 2장에서는 관련연구를 다루었고 3장에서는 이동객체의 TP 최근접 쌍 질의의 처리 알고리즘을 제안하고, 4장에서는 결론을 맺도록 한다.

2. 관련 연구

2.1 TPR-tree

이 논문은 2003년도 두뇌한국21사업에 의하여 지원되었음.

TPR-tree는 R-tree를 기반으로 움직이는 객체들의 미래 위치를 검색하는 시공간 질의를 지원하는 멀티웨이 인덱스 구조이다[2]. TPR-tree는 R-tree 기반에 이동객체의 속도정보를 추가한 구조이다. 삼입 삭제는 R*-tree와 비슷하다[3]. 엔트리의 단말 노드에는 이동객체의 위치정보와 속도정보를 담고 있으며 중간 노드는 하위 노드를 포함하는 보수적인 MBR (Conservative MBR)이다. 보수적인 MBR이란 포함한 속도가 가장 작은 객체의 속도를 자신의 하계 속도로 하고, 가장 큰 속도를 가지는 이동객체의 속도를 자신의 상계 속도로 한다. 이를 통해 시간이 흘러도 모든 이동객체를 여전히 포함하고 있게 된다. 결국 보수적인 MBR은 업데이트 될 때까지 줄어들지 않고 재구성(rebuild)을 통해서만이 작아질 수 있다.



(a) 시간이 0일 때 속도를 가진 MBR (b) 시간이 1일 때의 MBR

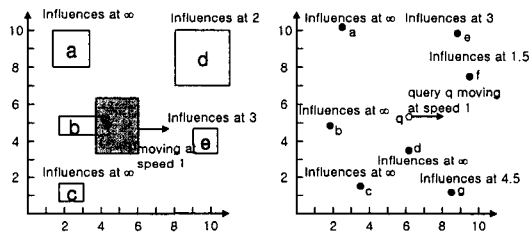
<그림 1> TPR-tree의 예

보수적 MBR을 이용함으로써 이동객체의 현재 위치를 인덱싱하고 미래 위치를 예측할 수 있다. 예를 들어 1차원이라고 할 때 어떤 이동객체 P가 t_0 시간 때의 위치와 속도가 (P_0, v) 일 때 미래 t 시간일 때의 위치는 $P_t = P_0 + v(t - t_0)$ 이다. 이와 같이 미래 어느 시각의 이동객체의 위치 정보를 알 수 있다. 그리고 업데이트 하면 과거의 정보는 남겨두지 않는다.

2.2 TP 질의(Time-Parameterized Queries)

TP 질의는 연속적인 시공간 질의를 의미한다. 이 질의는 현재의 질의 결과뿐만 아니라, 현재 질의의 유효시간 그리고 유효시간 후에 바뀌는 질의 결과를 사용자에게 반환한다. TP 질의에는 TP 범위 질의(TP window query), TP 최근접 이웃 질의(TP NN query), TP 조인 질의(TP Join query), TP 최근접 쌍 질의(TP closest pair query) 등이 있다.

그림 2(a)는 TP 범위 질의의 예를 보여준다. 질의 영역이 속도 1로 오른쪽으로 이동한다면 시간 1 이후에 b는 더 이상 질의 영역에 존재하지 않고 질의를 만족하는 객체는 없다. 그러므로 <b,1,null>를 사용자에게 반환해 준다. 또, 그림 2(b)는 최근접 이웃 질의의 예를 보여준다. 질의 지점이 오른쪽으로 1의 속도로 이동하고 있을 때, 현재 질의 결과는 d가 되지만 유효 시간(시간 1)이 지나면 질의 결과가 f로 바뀐다. 그림 2에서 볼 수 있듯이 시간 경계 간격을 갖는 질의는 이동객체들의 정확성이 시간에 따라 영향을 받아 질의 결과가 바뀐다.



(a) 시간 경계간격을 갖는 범위 질의의 예 (b) 시간 경계간격을 갖는 최근접 이웃 질의의 예

<그림 2> TP 질의의 예

이와 같은 시간 경계 간격을 갖는 다양한 질의들에 대해 연구가 이루어져 왔지만, 시간 경계 간격을 갖는 최근접 쌍 질의는 다루어지지 않았다.

3. TP 최근접 쌍 질의의 처리

이동객체의 공간 질의의 결과는 그렇게 유용하지 않다. 이동 객체나 질의가 이동하면 질의 결과도 따라서 변할 수 있기 때문이다. 사용자로부터 TP 질의를 받으면 질의를 수행하여 사용자에게 <R,T,C>를 반환한다. R은 현재의 질의 결과이고, T는 현재 질의 결과의 유효한 시간이며, C는 그 유효한 시간 후에 바뀐 질의 결과이다.

3.1 TP 최근접 쌍 질의의 정의

두 데이터 집합 $P(p_1, p_2, \dots, p_n), Q(q_1, q_2, \dots, q_n)$ 가 있을 때 P와 Q의 데이터 중에서 거리가 가장 짧은 쌍이 최근접 쌍이다. 이 최근접 쌍 (p_m, q_n) 은 아래의 수식을 만족한다.

$$dist(p_i, q_i) \geq dist(p_m, q_n), \forall p_i \in P, q_i \in Q$$

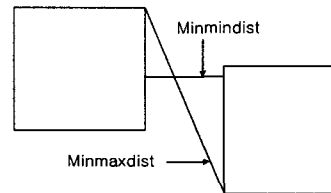
TP 최근접 쌍 질의는 현재의 최근접 쌍과 이 질의 결과의 유효 시간, 그리고 유효시간이 지난후의 질의 결과를 사용자에게 반환 해준다.

3.2 TP 최근접 쌍 질의 처리 알고리즘

TP 최근접 쌍 질의를 처리하는 것은 두 가지 단계로 이루어진다. 그것은 현재 질의 결과를 찾는 단계와, 현재 질의 결과의 유효시간을 계산하는 단계이다.

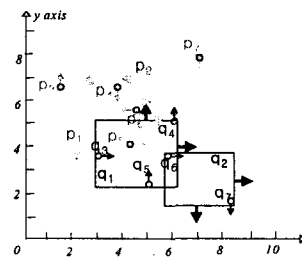
3.2.1 현재 질의 결과의 검색

현재의 질의 결과는 힙프 알고리즘을 사용하여 구한다 [4]. 이 알고리즘은 MinMaxDist와 MinMinDist 두개의 측정기준을 사용한다. 그림 3에서 보이는 바와 같이 MinMaxDist는 가장 가까운 변들 사이의 최대 거리이고, MinMinDist는 두 사각형 사이의 최소 거리이다.



<그림 3> 두 개의 측정 기준

- 1) 두 개의 TPR-tree의 루트 노드로부터 시작하여, D를 ∞ 로 설정하고 힙프를 초기화 한다.
- 2) 만약 두개의 중간 노드를 액세스 했다면, 모든 가능한 MBR 쌍의 MinMaxDist를 계산한다. ① 만약 MinMaxDis의 최저 값이 D보다 작다면 D를 업데이트하고, 모든 가능한 쌍들의 MinMinDist를 계산한다. ② 그 중에서 MinMinDist가 D보다 작은 쌍들을 힙프에 넣는다. ③ 거리가 가장 작은 단말노드 쌍을 저장한다.
- 3) 만약 단말노드를 액세스하면 모든 가능한 포인트 쌍들의 거리를 계산한다. 그 중에서 가장 작은 거리가 D보다 작으면 D를 업데이트한다.
- 4) 만약 힙프가 비었으면 알고리즘을 중지한다.



Heap		Result
1	(p ₂ , q ₁)	(p ₂ , q ₁)
2	(p ₂ , q ₂)	(p ₃ , q ₃)
3	(p ₃ , q ₃)	(p ₃ , q ₃)

<그림 3> 현재 질의 결과

- 5) 다음 힙프의 맨 위에 있는 쌍을 액세스한다. 만약 그 쌍의 MinMinDist가 D보다 크면 알고리즘을 중지하고, 그렇지 않으면 알고리즘 2)를 적용한다.

그림 4는 현재 최근접 쌍을 찾는 알고리즘의 예이다.

3.2.2 유효시간의 계산

두개의 이동객체 집합 $P(p_1, p_2, \dots, p_n)$ 와 $Q(q_1, q_2, \dots, q_n)$ 에서 (p_{cp}, q_{cp}) 가 현재의 최근접 쌍의 질의 결과라고 하고, V_i, p_{cp} 는 P_{cp} 의 i 차원의 속도이고, p_i, q_i 는 P와 Q의 데이터의 i 차원의 위치정보이다. p_{cpi}, q_{cpi} 는 P_{cp} 와 Q_{cp} 의 i 차원의 위치정보이다. 이때 아래의 수식을 만족하는 T가 현재 질의의 유효시간이 된다.

$$|p_{cp}(t), q_{cp}(t)| \geq |p(t), q(t)|, t \geq 0$$

이 수식을 정리하면 $At^2 + Bt + c \leq 0, (t \geq 0)$ 이런 2차 부등식이 나온다.

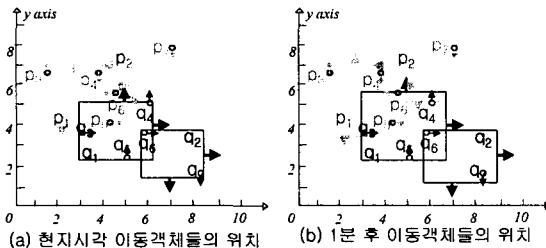
$$A = \sum_{i=1}^n [(V_i \cdot p - V_i \cdot q)^2 - (V_i \cdot p_{cp} - V_i \cdot q_{cp})^2]$$

$$C = \sum_{i=1}^n [(p_i - q_i)^2 - (p_{cpi} - q_{cpi})^2]$$

$$B = \sum_{i=1}^n 2[p_i - q_i](V_i \cdot p - V_i \cdot q) - (p_{cpi} - q_{cpi})(V_i \cdot p_{cp} - V_i \cdot q_{cp})$$

위 수식을 이용해서 우리는 유효시간 T를 구할 수 있다. 동시에 이 유효시간이 지난 후에 바뀌게 되는 질의 결과도 찾아낸다. 알고리즘은 아래와 같다.

- 1) 현재 질의 처리 알고리즘에서 계산한 거리가 가장 작은 단말노드 쌍의 각 포인트 쌍들의 유효시간을 계산해서 가장 작은 유효시간을 T_{min} 로 설정하고, 거리를 D_{min} 이라 한다.
- 2) T_{min} 시간 후의 각 단말 노드 쌍들의 거리를 계산한다. 그 중에서 단말 노드 사이의 거리가 D_{min} 보다 작은 모든 단말노드 쌍들을 히프에 넣는다.



	Heap			Result
1	(p ₂ , q ₁)	(p ₅ , q ₅)	(p ₂ , q ₂)	Empty
2	(p ₅ , q ₅)	(p ₂ , q ₂)		Empty
3	(p ₅ , q ₅)			(p ₅ , q ₅)

(c) 질의 결과

<그림 5> 유효시간 계산하는 예

- 3) 히프의 맨 위에 있는 쌍을 액세스한다. 만약 그 쌍에서 계산한 유효시간이 T_{min} 보다 작으면 T_{min} 를 업데이트하고, T_{min} 시간 후의 최근접 쌍의 거리를 계산한다. ① 만약 다음 단말 노드 쌍의 거리가 최근접 쌍의 거리보다 크면 알고리즘을 중지한다. ② 만약 거리가

작으면 알고리즘 3)을 다시 적용한다.

- 4) 히프가 비면 알고리즘을 중지한다.

그림 5는 이 알고리즘의 간단한 예를 보여준다. 먼저 현재 시각의 노드들의 거리를 계산하여 히프에 넣는다. 다음 히프의 맨 앞에 있는 단말 노드를 액세스하여 가장 작은 유효시간을 계산한 후, T_{min} 를 그 유효시간으로 업데이트 한다. 다음 T_{min} 시간 후의 MBR 쌍들의 거리를 계산하여 최근접 쌍의 거리보다 작은 쌍들을 히프에 넣는다. 이렇게 히프가 비거나 MBR 쌍의 거리가 가장 인접한 쌍의 거리보다 크면 알고리즘을 종료한다.

4. 결론

이동객체는 시간에 따라 그 위치가 변하므로 그 질의 결과는 유효시간을 갖는다. 일반적인 공간 질의는 이동객체의 질의에 대해서는 유효시간을 명시해주지 못하고 있다. 본 논문에서는 최초로 TP 최근접 쌍 질의를 처리하는 알고리즘을 제시하였다.

이 알고리즘은 먼저 히프 알고리즘을 사용하여 현재의 최근접 쌍을 찾아내고, 다음 현재의 질의 결과로부터 현재 질의 결과의 유효시간을 계산하는 방법을 제시하였다. 즉 질의 결과가 바뀌는 시각과 바뀐 질의 결과를 찾아낸다. 이 알고리즘을 이용하여 최근접 쌍 질의를 던지는 사용자들에게 더 많은 정보를 제공해준다. TP 최근접 쌍 질의는 사용자가 질의를 주면 현재의 최근접 쌍, 그리고 현재의 질의 결과의 유효시간과 이 유효시간 후에 바뀌게 되는 질의 결과를 사용자에게 반환함으로써, 사용자들에게 더욱 도움이 될 것이다.

참고문헌

- [1] Yufei Tao, Dimitris Papadias, Time-Parameterized Queries in Spatio-Temporal Databases. SIGMOD, 2002.
- [2] Simonas Saltenis, Christian S. Jensen, Scott T. Leutenegger, Mario A. Lopez, Indexing the Positions of Continuously Moving Objects. SIGMOD, 2000.
- [3] N. Beckmann, H.P. Kriegel, R. Schneider and B. Seeger, The R*-tree: an Efficient and Robust Access Method for Points and Rectangles. SIGMOD 1990.
- [4] A. Corral, Y. Manolopoulos, Y. Theodoridis, M. Vassilakopoulos, Closest Pair Queries in Spatial Databases, Corral 2000.
- [5] Agarwal, P., Arge, L., Erickson, J., Indexing Moving Points. PODS, 2000.
- [6] Hoda Mokhtar, Jianwen Su, Oscar Ibarra, On Moving Object Queries, PODS, 2002.
- [7] Jun Zhang, Manli Zhu, Dimitris Papadias, Yufei Tao, Dik Lun Lee, Location-based Spatial Queries. SIGMOD 2003.
- [8] Yufei Tao, Dimitris Papadias, Jimeng Sun, The TPR*-Tree: An Optimized Spatio-Temporal Access Method for Predictive Queries. VLDB 2003
- [9] A. Gattman, R-tree: A Dynamic Index Structure for Spatial Searching. SIGMOD 1984.