

도로네트워크 기반 이동 객체들 간의 배타적 최근접 쌍 모니터링 방법

권 오 제* · 백 윤 선** · 이 기 준***

Continuous Monitoring of k-Exclusive Closest Pairs in Road Network

Oje Kwon* · Yun-Sun Baek** · Ki-Joune Li***

요 약

배타적 최근접 쌍 검색 방법은 가까이 있는 승객과 택시 간에 쌍을 맺어주는 것과 같은 실 생활 응용 분야에 유용하게 쓰일 수 있으나, 그 중요성에 비해 많은 연구가 진행되지 않았다. 두 이동 객체가 배타적으로 최근접 쌍이 되기 위해서는 객체는 반드시 하나의 결과 쌍에 속해야 한다. 하지만, 이동 객체는 위치가 빈번하게 변하기 때문에 초기의 배타적 최근접 쌍 결과를 모니터링 해 줄 필요가 있다. 본 논문에서는 도로 네트워크를 기반으로 k 개의 배타적 최근접 쌍을 구하고 그 결과를 모니터링 하기 위한 방법을 제안한다. 제안하는 방법은 모든 객체에 대해 배타적 최근접 쌍 검색을 재수행하지 않고, 결과가 갱신되는 범위를 제한하여 부분적인 재수행을 함으로써 모니터링 수행 시간을 줄였다. 다양한 실제 도로 네트워크를 이용한 실험들을 통해 본 논문에서 제안하는 모니터링 방법의 성능을 측정하였다. 실험 결과는 본 논문에서 제안하는 모니터링 방법의 정확도가 일반적인 일괄처리 방법들에 비해 뛰어난 것을 보인다.

주요어 : 배타적 최근접 쌍, 결과 모니터링, 이동 객체, 도로 네트워크

ABSTRACT : Finding exclusive closest pairs in road network is very useful to real applications such as, for example, finding a closest pair between a passenger and a nearby taxi in a road network. Few studies, however, have been interested in this problem. To match two close moving objects exclusively, one object must belong to only one result pair. Because moving objects in a road network change their position continuously, it is necessary to monitor closest pair results. In this paper, we propose a methodology to monitor k exclusive closest pairs via a road network. Proposed method only updates the results which are influenced by objects' movement. We evaluated the performance of the proposed method with various real road network data. The results show that our method produces better accuracy than normal batch processing methods.

Keywords : Exclusive closest pair, Monitoring exclusive closest pair, Moving object, Road network

1. 서 론

위치 기반 서비스에 대한 관심이 높아지면서 도로 위의 이동 객체들의 위치 정보를 이용한 공간 질의 처리에 관한 연구들이 많이 진행되고 있다. 이중 이동 객체들 간의 배타적인 최근접 쌍을 찾는 연구의

중요성이 증가하고 있다. 배타적 최근접 쌍은 이동 객체들이 서로 배타적으로 최근접 쌍을 맺는 것으로 하나의 이동 객체는 반드시 하나의 최근접 쌍에만 포함된다. 예를 들면, 승객에게 가장 가까운 택시를 할당해주는 서비스가 있다고 가정하자. 서비스 서버는 서비스를 요청한 승객들의 위치 정보와 보유한

*부산대학교 컴퓨터공학과 박사과정

**삼성전자 연구원

***주저자, 부산대학교 정보컴퓨터공학부 교수

택시들의 위치 정보를 GPS 등의 디바이스들을 통해 획득하여 택시와 승객간의 최근접 쌍을 질의하여 승객에게 가장 가까운 택시를 할당해준다. 이 때, 승객은 여러 대의 택시를 요청받을 필요가 없고 택시 또한 여러 승객을 태울 수 없기 때문에 승객과 택시는 배타적으로 쌍이 맺어지게 된다.

이동 객체는 시간에 따라 움직이기 때문에 이전의 결과가 시간이 흐름에 따라 계속해서 유효할 수 없다. 예를 들어, 콜택시 서비스의 예의 경우, 현재 서비스를 요청하였을 때 맺어지는 승객-택시 쌍의 결과가 시간이 흐른 뒤 계속해서 유지될 수는 없다. 처음에 승객1과 택시1이 쌍으로 맺어졌지만 택시1이 승객1의 방향과 멀어지는 방향으로 움직이거나 택시1이 교차로를 지나치면서 승객1의 위치에 도달하는 거리가 멀어지는 경우 콜택시 서비스는 택시1보다 승객1에 더 가까운 다른 택시2를 할당해주어야 한다. 따라서 시간에 따라 질의 결과를 계속해서 모니터링 해주어야 한다.

질의 결과를 모니터링하기 위해 최근접 쌍 질의를 매시간 재수행할 수 없다. 다수의 이동 객체에 대해 최근접 쌍 질의를 수행하는데 부하가 크고 질의 주체와 대상이 모두 이동 객체이기 때문에 빈번한 위치 변화로 인해 질의 결과의 갱신이 자주 일어나기 때문이다. 따라서 질의를 재수행하지 않고 부분적인 갱신을 통해 질의 결과를 갱신하기 위한 방법이 필요하다.

본 논문에서는 도로 네트워크를 기반으로 배타적인 최근접 쌍 결과를 모니터링하기 위한 방법을 제안한다. 모니터링 방법은 전체 도로 네트워크를 검색하지 않고 결과값 갱신에 영향을 받은 이동 객체를 중심으로 검색 영역을 제한한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 연구들에 대해 살펴보고 본 논문의 연구 동기에 대해 서술한다. 3장에서는 배타적 최근접 쌍을 정의하고 이를 검색하기 위한 방법론을 설명한다. 4장에서는 3장에서 서술한 방법으로 찾은 배타적 최근접 쌍 결과를 모니터링 하기 위한 방법론을 설명한다. 5장에서 다양한 실험을 통해 본 논문에서 제안하는 방법론들의 성능 평가를 수행하고 마지막으로 6장에서 논문을 마무리한다.

2. 관련 연구 및 연구 동기

이동 객체는 시간이 흐름에 따라 위치가 변하기 때문에 초기에 찾은 최근접 쌍 결과는 시간의 흐름에 따라 갱신되어질 수 있다. Yu et al.(2005)는 k 최근접 질의를 모니터링하기 위해 그리드를 기반으로 한 이동 객체 기반 색인과 질의점 기반 색인을 제안하였다. 지정희 et. al(2004)는 이동 객체 궤적 기반의 연속 최근접 질의 방법을 제안하였다. Mouratidis et al. (2006)는 도로 네트워크를 기반으로 한 k 최근접 질의를 모니터링하기 위한 방법을 제안하였다. 이 방법은 질의점을 기준으로 k 번째의 최근접 객체까지의 거리에 포함되는 간선들을 대상으로 이동객체(질의점, 질의대상)가 움직일 때 최근접 결과를 모니터링한다.

Zhu et al.(2006)는 이동 객체마다 위치가 갱신되어도 이전 최근접 쌍의 결과에 영향을 주지 않는 거리인 안전 거리(safe distance)를 정의하고 안전 거리를 벗어난 경우만 서버에서 최근접 쌍 결과를 갱신하는 방법을 제안하였다. Hou et al.(2008)는 CPM(Kyriakos et al., 2005)을 이용한 이동 객체의 최근접쌍 모니터링 방법을 제안하였다. 이 방법은 각각의 질의점에 대해 가까운 순서대로 이동 객체를 나누어진 셀에 저장함으로써 질의 처리 시 이동 객체에 대한 불필요한 접근을 피한다. 그밖에 이동 객체의 위치 예측을 통한 질의 처리 방법도 연구되었다(홍동숙 et. al, 2007).

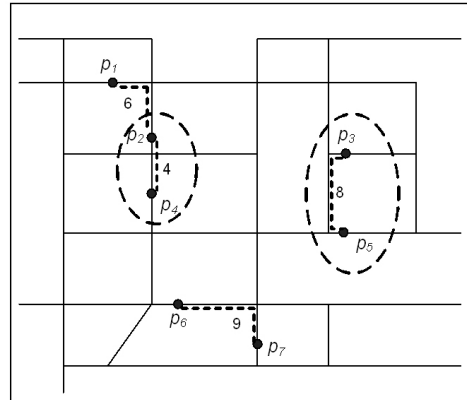
하지만 지금까지 연구된 방법들은 유클리디안 네트워크를 기반으로 한다. 도로 네트워크에서의 거리는 유클리디안 거리와는 다르다. 따라서 도로 네트워크 거리를 이용하여 배타적 최근접 쌍을 찾기 위한 방법이 필요하다. 뿐 만 아니라, 질의 대상과 질의점 모두 이동 객체인 경우를 고려하고 있지 않다. 본 논문에서는 도로 네트워크 거리를 이용하여 이동 객체들 간의 k 개의 배타적 최근접 쌍을 찾고, 결과를 모니터링하기 위한 방법론을 제안한다.

3. 도로 네트워크 기반 배타적 최근접 쌍 검색 방법

본 장에서는 k 개의 배타적 최근접 쌍을 찾기 위한 방법으로 클러스터링 방법을 이용한 클러스터링 기

<표 1> 기호 정의

기호	설명
P	이동객체 집단
p, q	P 에 속하는 임의의 이동객체
p_N	p 의 최근접 객체
p_i, p_j	최근접 쌍을 맺는 두 이동객체
$dist(p,q)$	p, q 의 도로 상의 거리
$ECP_k(P)$	P 에서 k 번째 배타적 최근접 쌍
$CP(P)$	P 에서 최근접 쌍
$R_k(P)$	P 에서 k 개의 배타적 최근접 쌍 집합



[그림 1] 배타적 최근접 쌍의 예

반 배타적 최근접 쌍 방법(k-Cluster based Exclusive Closest Pair, k-CECP)을 기술한다. 이에 앞서, 본 논문에서는 도로 네트워크에서의 최근접 쌍을 정의 1과 같이 정의한다. <표 1>은 정의에 사용되는 기호들을 나타낸다.

정의 1. 최근접 쌍(Closest Pair, CP)

이동 객체 집합 P 에 속하는 모든 이동 객체 p, q 에 대해, 최근접 쌍은 다음과 같이 정의된다.

$$\forall p, q \in P,$$

$$CP(P) = \{(p_i, p_j) | dist(p_i, p_j) = \min(\{dist(p, q)\})\}$$

도로 네트워크에서 최근접 쌍은 도로 네트워크 상의 거리로 가장 가까운 두 객체 쌍을 의미한다. 본 논문에서는 정의 1을 바탕으로 k 개의 배타적 최근접 쌍을 정의 2와 같이 정의한다.

정의 2. 배타적 최근접 쌍(Exclusive Closest Pair, ECP)

이동 객체 집합 P 에 속하는 모든 이동 객체 p 에 대해, 배타적 최근접 쌍은 다음과 같이 정의된다.

$$\forall p \in P,$$

$$ECP_1(P) = CP(P),$$

$$ECP_k(P) = ECP_{k-1}(P) \cup CP(P - R_{k-1}(P)),$$

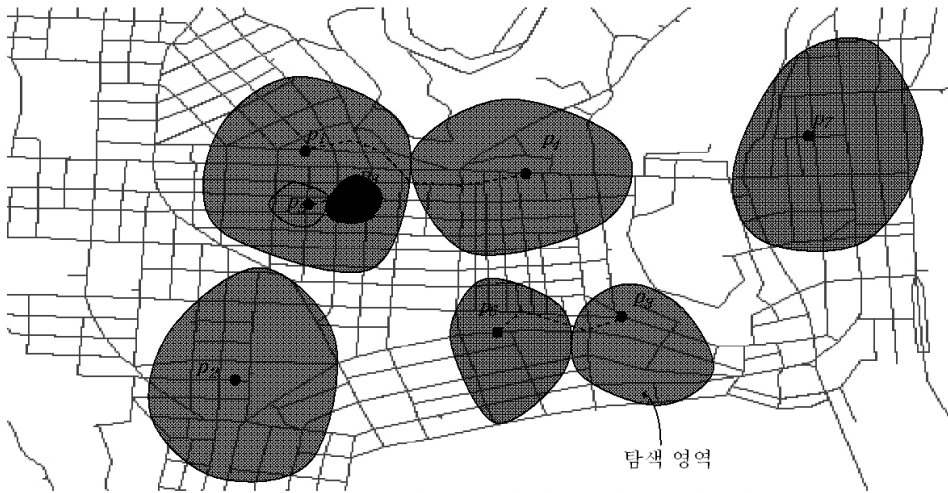
$$R_k(P) = \{p_i | \exists q, (p_i, q) \in ECP_k(P)\} \cup \{p_j | \exists q, (q, p_j) \in ECP_k(P)\}$$

배타적 최근접 쌍에서 배타적이란 하나의 최근접 쌍을 맺고 있는 이동 객체는 또 다른 이동 객체와

최근접 쌍을 맺을 수 없음을 의미한다. [그림 1]은 2개의 배타적 최근접 쌍을 구하는 예를 보여준다. p_1 와 p_2 는 정의1에 따라 두 이동 객체 간의 거리가 가장 짧기 때문에 첫 번째 최근접 쌍이 된다. 두 번째 최근접 쌍은 첫 번째 최근접 쌍 다음으로 거리가 가까운 p_1 과 p_2 이다. 하지만 p_2 는 이미 첫 번째 최근접 쌍에 포함되었기 때문에 정의2에 따라 배타적 최근접 쌍에 속할 수 없다. 따라서 첫 번째 최근접 쌍에 속한 p_2 와 p_1 를 제외한 나머지 이동객체들 중 가장 거리가 짧은 p_3, p_5 가 두 번째 배타적 최근접 쌍이 된다.

앞서 정의한 배타적 최근접 쌍을 찾기 위해서, 본 논문에서는 클러스터링 기반 배타적 최근접 쌍 방법을 제안한다. 이 방법은 [그림 2]와 같이 도로 네트워크 위의 이동 객체들을 중심으로 탐색 영역을 확장해가며 최근접 쌍을 찾는다. 클러스터 기반 배타적 최근접 쌍 방법은 최대 도로 네트워크를 한번 검색함으로써 k 개의 최근접 쌍을 찾을 수 있다. 클러스터 기반 배타적 최근접 쌍 방법은 <표 2>의 자료구조를 필요로 한다.

클러스터링 기반 배타적 최근접 쌍 검색 방법은 2단계로 이루어진다. 첫 번째 단계는 <표 2>의 세 가지 자료구조에 대한 초기화 단계다. 두 번째 단계는 초기화된 자료구조를 가지고 최근접 쌍을 찾기 위해 도로네트워크 탐색 영역을 확장하는 단계다. 클러스터 기반 배타적 최근접 쌍 방법은 도로 네트워크의 노드를 중심으로 인접 노드로 탐색 범위를 넓혀가며 최근접 쌍을 찾는다. 초기화 단계에서는 각 노드로부터 인접한 간선 위에 존재하는 이동 객체들 간의

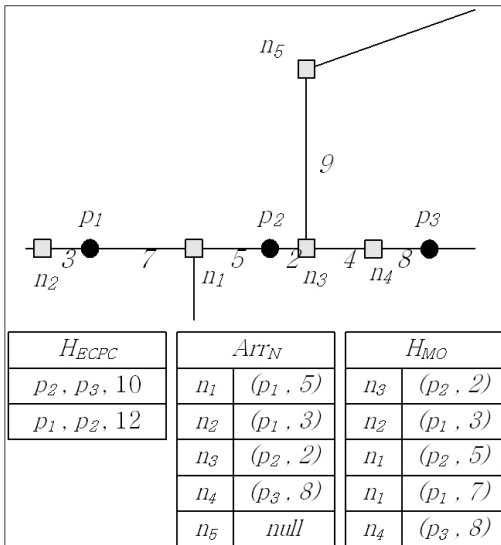


3-배타적 최근접 쌍 결과 : $(p_5, p_3), (p_6, p_3), (p_1, p_4)$

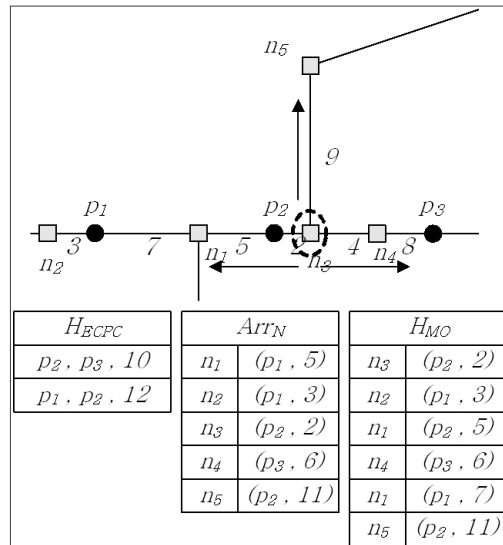
[그림 2] 클러스터 기반 배타적 최근접 쌍 검색 방법의 예

<표 2> k-CECP를 위한 자료구조

자료구조	설명
H_{MO}	도로 네트워크의 각 노드와 노드가 공유하는 간선 위의 이동 객체 간의 거리를 저장하는 힙
Arr_N	도로 네트워크의 각 노드와 노드가 공유하는 간선 위의 최근접 이동 객체 간의 거리를 저장하는 배열.
H_{ECPC}	배타적 최근접 쌍의 후보가 되는 두 객체와 두 객체 간의 거리를 저장하는 힙



(a) 초기화 단계



(b) 탐색 영역 확장 과정

[그림 3] 클러스터 기반 배타적 검색 방법 수행 과정

위치 정보를 이용하여 세 가지 자료구조를 초기화 한다.

초기화 단계가 끝나면 k개의 배타적 최근접 쌍을 찾기 위해 탐색 영역을 확장하는 과정이 수행된다.

탐색 영역 확장 과정은 노드와 이동 객체간의 거리가 가장 짧은 노드에서부터 시작된다. 시작 노드 n_s 는 인접한 간선을 공유하는 인접 노드 n_{adj} 까지 탐색 영역을 확장한다. 이 때 탐색 영역 내에서 두 노드의 관계에 따라 세 가지로 나누어 처리된다.

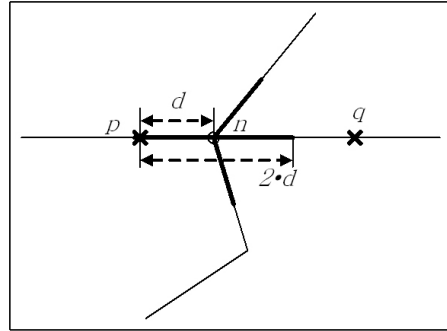
- 경우 1. n_{adj} 의 최근접 이동 객체를 알 수 없을 경우 또는, n_{adj} 의 최근접 이동 객체가 이미 최근접 쌍을 맺고 있을 경우 - [그림 3]-(b)에서 노드 n_2 가 (1)의 방향으로 탐색 영역을 확장할 때 인접 노드인 n_5 는 최근접 이동 객체 정보가 없다. 따라서 n_5 는 n_2 의 최근접 이동 객체인 p_2 의 아이디와 p_2 까지의 거리 정보를 가지고 이 값은 Arr_N 에 저장된다. 그리고 다음 탐색 영역 확장을 위해 $(n_5, p_2, 11)$ 정보가 H_{MO} 에 삽입된다.
- 경우 2. n_{adj} 와 n_s 의 최근접 이동 객체가 같을 경우 - [그림 3]-(b)에서 노드 n_2 가 (2)의 방향으로 탐색 영역을 확장할 때 인접 노드인 n_1 은 n_2 와 동일한 최근접 이동 객체 p_2 정보를 가지고 있다. 이 경우 n_1 은 이미 최근접 이동 객체의 정보를 가지고 있으므로 추가 작업이 수행되지 않는다.
- 경우 3. n_{adj} 와 n_s 의 최근접 이동 객체가 다른 경우 - [그림 3]-(b)에서 노드 n_2 가 (3)의 방향으로 탐색 영역을 확장할 때 인접 노드인 n_4 는 자신의 최근접 이동 객체 p_3 까지의 거리 보다 n_2 의 최근접 이동 객체 p_2 까지의 거리가 더 가깝다. 따라서 Arr_N 내의 n_4 의 최근접 이동 객체 정보는 p_2 로 갱신된다. 그리고 탐색 영역 확장을 위해 $(n_4, p_2, 6)$ 의 정보가 H_{MO} 에 삽입된다. 경우 3에서는 두 이동 객체 p_2 와 p_3 의 거리가 계산된다. 이 때 p_2, p_3 의 거리는 최단 거리를 보장하지 못하기 때문에 거리 정보는 후보 집합 H_{ECPC} 에 저장된다.

H_{ECPC} 에 포함된 후보객체 쌍은 다음 정리 1, 2를 만족할 경우 최근접 쌍이 된다.

정리 1. 노드 n 과 n 의 최근접 이동 객체 p 까지의 거리를 d 라 할 때, 노드 n 을 중심으로 반지름 d 의 범위 내에 p 와의 거리가 $2 \bullet d$ 보다 짧은 이동 객체 q 는 존재하지 않는다.

[그림 4]와 같이 노드 n 에 인접한 간선 위에서 최

대 d 범위 내에는 이동 객체가 p 밖에 존재하지 않는다. 즉, 이동 객체 p 로부터 $2 \bullet d$ 범위 내에 다른 이동 객체는 존재하지 않는다. 정리 1로부터 정리 2를 유도할 수 있다.



[그림 4] 노드와 최근접 객체 간의 거리

정리 2. H_{ECPC} 에 포함된 모든 후보 쌍들에 대해 다음을 만족하는 쌍 (p, q) 는 배타적 최근접 쌍이 된다.

$$\forall (p, q) \in H_{ECPC}, (p, q) \text{ is Closest Pair} \\ \text{iff } Dist_{RN}(p, q) < 2 \bullet H_{MO}.mindist$$

탐색영역의 확장 과정은 H_{MO} 에 속해있는 노드들 중 최근접 이동 객체까지의 거리가 짧은 노드부터 진행된다. 정리 1에 의해 다른 노드에서도 H_{MO} 의 최단 거리 $H_{MO}.mindist$ 보다 짧은 거리의 이동 객체 쌍은 존재하지 않는다. 이동 객체 간의 거리는 노드와의 거리로부터 점진적으로 구해지기 때문에 탐색 영역 확장 과정에서 구해진 이동 객체간의 거리가 현재의 H_{MO} 의 최단 거리 두 배보다 짧으면 이는 최단거리가 된다. 이동 객체 쌍이 정리 2를 만족하면, 해당 이동 객체 쌍은 최근접 쌍이 되고 결과 집합 R 에 저장된다. 배타적 최근접 쌍 탐색 과정은 집합 R 의 개수가 k 가 될 때까지 반복된다.

4. 도로 네트워크 기반 배타적 최근접 쌍 결과 모니터링 방법

이동 객체의 위치 변화가 최근접 쌍의 결과에 영향을 미치는 경우는 크게 (1) 최근접 쌍 결과에 포함된

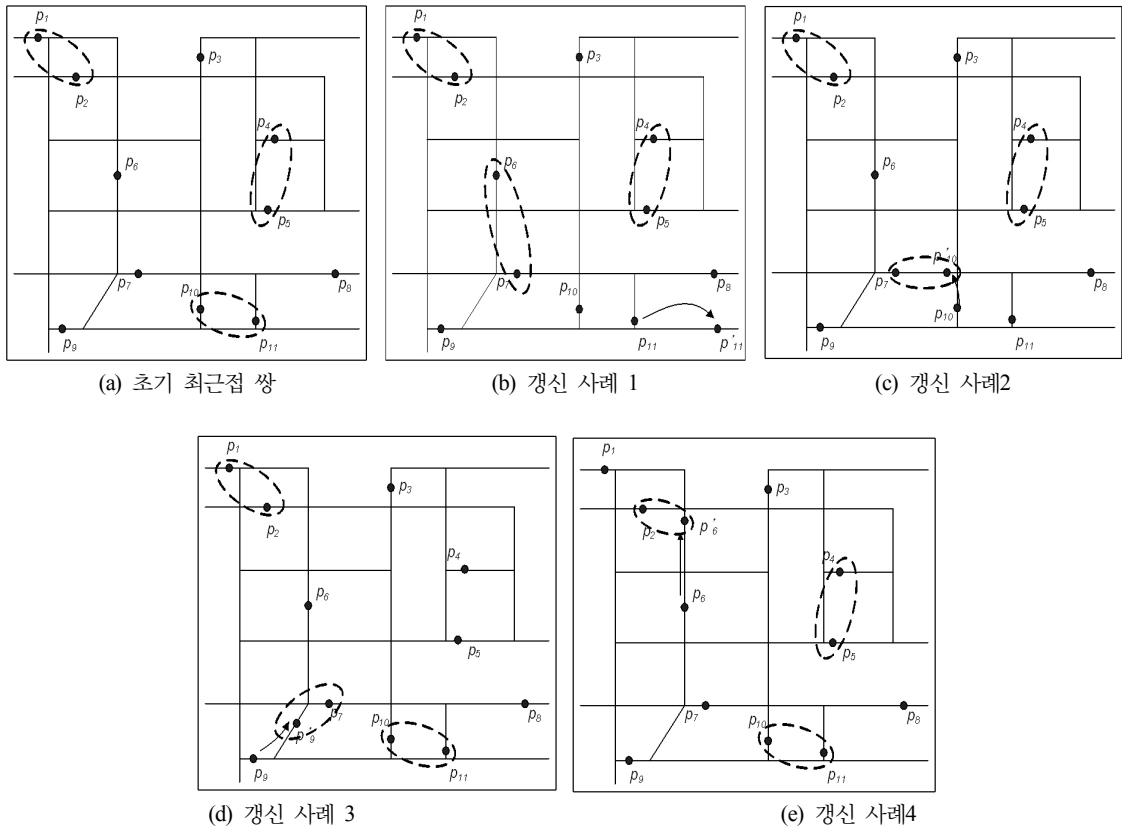
이동 객체의 움직임에 의한 결과 변화와 (2) 최근접 쌍 결과에 포함되지 않은 이동 객체의 움직임에 의한 결과 변화로 나뉜다. 이를 정리해 보면 다음과 같다.

- 경우 1. 최근접 쌍을 맺고 있는 이동 객체가 위치 이동 후 현재 최근접 쌍이 깨어지는 경우 - [그림 5](b)
- 경우 2. 최근접 쌍을 맺고 있는 이동 객체가 위치 이동 후 다른 이동 객체와 최근접 쌍을 이루는 경우 - [그림 5](c)
- 경우 3. 최근접 쌍에 포함되지 않은 이동 객체가 위치 이동 후 최근접 쌍을 맺는 경우 - [그림 5](d)
- 경우 4. 최근접 쌍에 포함되지 않은 이동 객체가 위치 이동 후 다른 최근접 쌍의 이동 객체와 최근접 쌍을 맺는 경우 - [그림 5](e)

변화가 최근접 쌍 결과에 영향을 미치는 경우를 순서대로 보여준다. [그림 5](a)와 같이 3개의 최근접 쌍이 존재한다고 할 때, 최근접 쌍 결과에 포함된 이동 객체 p_{11} 이 위치 변화를 하여 p_{11} 이 속한 최근접 쌍의 거리가 길면 p_{11} 이 속한 최근접 쌍 (p_{10}, p_{11})은 결과 집합에서 삭제되고 그 보다 거리가 더 짧은 새로운 최근접 쌍 (p_6, p_7)이 결과 집합에 삽입된다([그림 5](b)). 만약 최근접 쌍 결과에 포함된 이동 객체 p_{10} 이 위치 변화를 하였을 때 원래의 최근접 객체 p_{11} 가 아닌 다른 이동 객체 p_7 와 최근접 쌍을 맺을 경우 원래의 최근접 쌍 (p_{10}, p_{11})은 결과에서 삭제되고 새로운 최근접 쌍 (p_{10}, p_7)이 결과에 삽입된다([그림 5](c)).

만약 최근접 쌍 결과에 포함되지 않은 이동 객체 p_9 가 위치 이동 후 맺게 된 최근접 쌍의 거리가 결과 쌍에 포함된 최근접 쌍의 거리보다 짧을 경우 새로운 최근접 쌍 (p_9, p_7)이 결과에 삽입되고 원래 결과에서

[그림 5]는 도로 네트워크에서 이동 객체의 위치



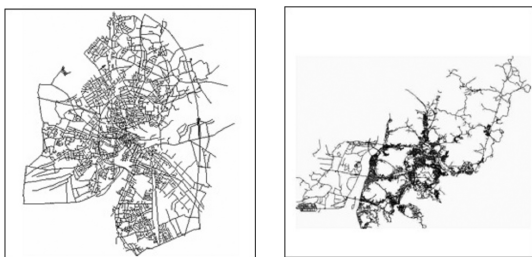
[그림 5] 이동 객체의 위치 변화에 따른 배타적 최근접 쌍 결과 갱신의 예

거리가 가장 긴 쌍 (p_4, p_5)는 삭제된다([그림 5]-(d)). 만약 최근접 쌍 결과에 포함되지 않은 이동 객체 p_6 이 위치 이동 후 결과에 포함된 최근접 쌍의 한 이동 객체 p_2 와 최근접 쌍을 맺는 경우 새롭게 맺어진 최근접 쌍 (p_6, p_2)의 거리가 원래의 최근접 쌍 (p_1, p_2)의 거리보다 짧다면 (p_1, p_2)는 결과에서 삭제되고 새로운 (p_6, p_2)이 결과에 삽입된다([그림 5]-(e)).

위치가 변한 이동 객체의 최근접 쌍 검색 과정은 클러스터링 기반 배타적 최근접 쌍 검색 방법을 활용한다. 초기에 k 개의 최근접 쌍을 검색 과정을 거친 후 세 개의 자료구조의 정보를 활용하여 도로 네트워크 검색 범위를 최소화 하여 최근접 쌍 검색을 수행한다. 만약 현재의 탐색 영역 임계값 θ 범위 내에서 최근접 쌍이 찾아질 경우 추가적인 도로 네트워크의 탐색 영역 확장없이 최근접 쌍을 구할 수 있다. 반면에 현재의 탐색 영역 임계값 θ 범위 내에서 최근접 쌍을 찾을 수 없는 경우 탐색 영역 확장을 수행한다. [그림 5]-(c)의 p_{11} 과 같이 원래 최근접 쌍에 속했는데 갱신 후 최근접 쌍에서 제외된 이동 객체들은 위치 변화를 한 이동 객체들과 동일하게 갱신 과정을 거친다. 이는 갱신 후 결과에서 제외된 이동 객체들 역시 새로운 최근접 쌍을 맺어 결과에 포함될 수 있기 때문이다.

5. 성능 평가

실험을 통하여 본 논문에서 제안하는 2가지 배타적 최근접 쌍 검색 방법과 모니터링 방법의 성능을 평가하였다. 성능 평가를 위해 [그림 6]의 2가지 실제 도로 네트워크를 사용하였다.



(a) 올덴버그 (b) 부산

[그림 6] 실제 도로 네트워크

본 논문에서 제안하는 두 가지 배타적 최근접 쌍 검색 방법은 도로 네트워크의 간선과 노드의 수에 영향을 많이 받는다. 따라서 상대적으로 간단한 도로([그림 6]-(a))와 복잡한 도로([그림 6]-(b))의 네트워크에서 실험을 수행하였다. 각 도로 네트워크를 구성하는 간선과 노드 수는 <표 3>에 나타나 있다.

실험을 위한 독립변수는 <표 4>와 같다. <표 4>의 굵은 값들은 실험의 기본값을 나타낸다. 실험 시 독립변수에 대한 특별한 언급이 없는 경우 그 독립변수들은 기본값을 가진다.

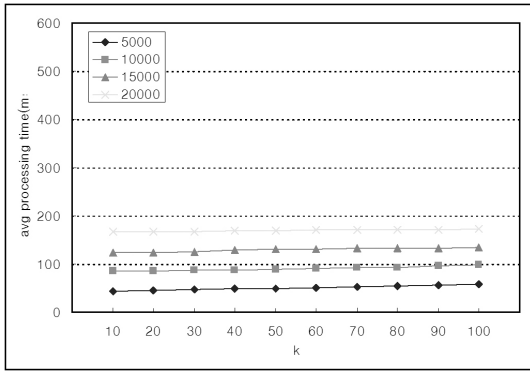
<표 3> 도로 네트워크의 노드와 간선 수

네트워크	노드 수	간선 수
올덴버그	5931	6970
부산	41882	49234

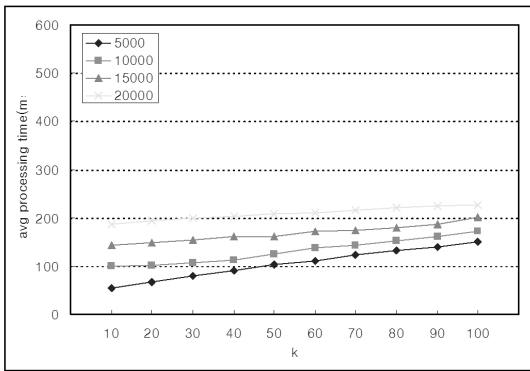
이동 객체들은 도로네트워크 위에 산재해서 분포한다고 가정한다. 이동 객체의 생성은 김보련 et al.(2005)의 도로 네트워크 이동 객체 생성기를 사용하였다. 이 생성기는 실제 교차로, 정체, 커브 등과 같은 실제 교통 상황을 반영하고 있고 이동 객체들의 속도 역시 실제 택시 및 자가용 데이터 분석을 통한 속도 모델을 바탕으로 변하기 때문에 실제에 가까운 이동 객체의 생성이 가능하다. 이동 객체는 GPS와 같은 소형 단말기를 이용해 자신의 위치 정보를 획득 가능하고 통신 단말기를 통해 1초 주기로 자신의 위치를 서버로 전송된다고 가정한다. 통신 시 발생하는 비용은 최소화되어 있다고 가정한다. 실험의 수행을 위해 윈도우XP 환경의 1GB의 메모리를 가진 팬티엄 4 3GHz PC를 사용하였다.

<표 4> 실험을 위한 독립변수

독립 변수	값
네트워크	올덴버그, 부산
이동객체 수	5000, 10000 , 15000, 20000
k	10, 20, 30, 40, 50 , 60, 70, 80, 90, 100
이동객체 평균속도(km/h)	30, 40, 50 , 60, 70, 80



(a) 올덴버그



(b) 부산

[그림 7] 도로 네트워크, 이동 객체 수 그리고 k 값의 변화에 따른 배타적 최근접 쌍 질의 처리 방법의 평균 수행 시간

5.1 배타적 최근접 쌍 검색 방법 성능 평가

[그림 7]은 도로 네트워크, 이동 객체의 수, 배타적 최근접 쌍의 수 k 값의 변화에 따른 클러스터 기반 배타적 최근접 쌍 검색 방법의 평균 수행 시간을 보여준다. 실험 결과 도로 네트워크의 크기가 클수록, 최근접 쌍의 수 k 가 클수록, 이동 객체의 수가 많을수록 수행 시간이 많음을 알 수 있었다.

5.2 배타적 최근접 쌍 모니터링 방법 성능 평가

본 논문에서 제안하는 배타적 최근접 쌍 모니터링 방법의 성능을 다음 두 가지 일괄 처리 방법과 비교하였다.

- 연속적인 일괄 처리 방식(Continuous Batch, CB) - 최

근접 쌍 방법이 수행된 후 이동 객체의 변경된 위치 정보를 반영하여 최근접 쌍 방법을 재수행한다.

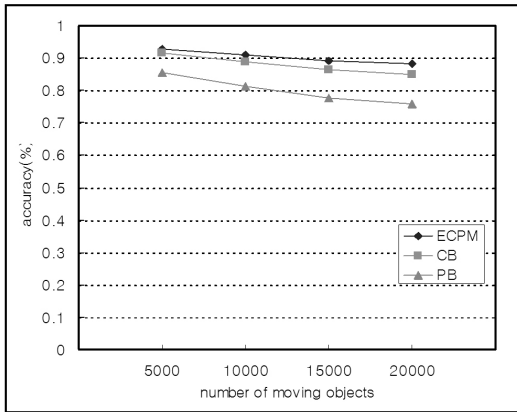
- 주기적인 일괄 처리 방식(Periodic Batch, PB) - 1초 주기로 클러스터 기반 최근접 쌍 검색 방법을 재수행한다.

본 논문에서는 질의 처리 결과가 얼마나 실제와 가까운지를 나타내는 정확도를 성능 척도로 사용한다. 모니터링 결과의 정확도는 실제 결과 집합을 R_s , 모니터링 된 결과 집합을 R_m 이라고 할 때 $\frac{|R_s \cap R_m|}{|R_s|}$ 로 나타낸다.

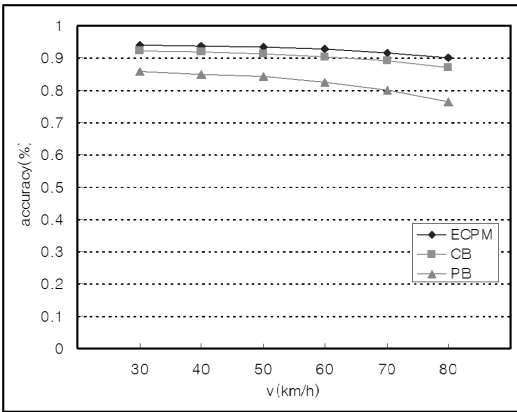
정확도는 이동 객체의 속도, 개수, 최근접 쌍의 수 그리고 모니터링 수행 시간에 영향을 받는다.

[그림 8](a)는 이동 객체의 수에 따른 정확도 변화를 나타낸다. 도로네트워크에 존재하는 이동 객체의 수가 많아지면 이동 객체들 간의 거리가 상대적으로 짧기 때문에 이동 객체의 움직임에 따른 최근접 쌍이 바뀔 확률이 증가한다. 따라서 이동 객체의 수가 많을수록 정확도 값은 작아진다. 주기적인 일괄 처리 방식은 주기적으로 모니터링을 수행하기 때문에 주기 내에 일어난 이동 객체의 위치 변화는 모니터링 수행에 반영할 수 없어 정확도가 낮다. 연속적인 일괄 처리 방식은 주기적인 일괄 처리 방식에 비해 정확도는 높으나 모니터링 수행 동안 발생하는 이동 객체의 위치 변화를 모니터링 수행에 반영할 수 없다. 본 논문에서 제안하는 모니터링 방법은 이동 객체의 위치 변화를 바로 결과 값에 반영하기 때문에 정확도가 다른 방법들에 비해 높고 이동 객체의 수 증가에 따른 정확도 감소 폭이 작다.

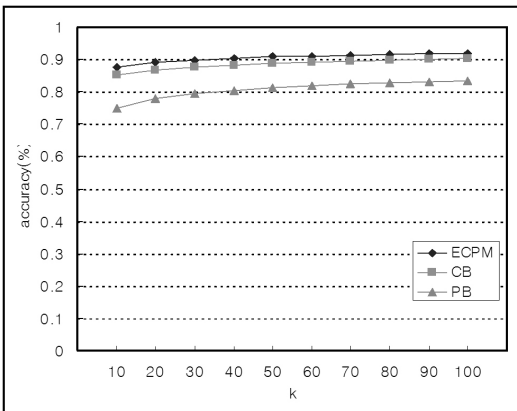
[그림 8](b)는 이동 객체의 속도에 따른 정확도의 변화를 나타낸다. 실험에서 이동 객체의 속도는 최고 속도를 나타낸다. 실험에 사용된 이동 객체 데이터의 속도는 최고 속도를 바탕으로 김보련 et. al(2005)의 데이터 생성기의 속도 모델에 의해 변화한다. 이동 객체의 속도가 빨라지면 이전의 최근접 쌍의 거리 변화가 커 최근접 쌍이 바뀔 확률이 높아지기 때문에 정확도 값은 떨어진다. 하지만 본 논문에서 제안하는 모니터링 방법은 이동 객체의 위치 변화를 바로 결과 값에 반영하기 때문에 정확도가 높게 유지되었다. [그림 8](c)는 배타적 최근접 쌍의 수 k 에 따른 정확도의 변화를 보여준다. 배타적 최근접 쌍의 개수가



(a) 이동 객체의 수



(b) 이동 객체의 속도



(c) k

[그림 8] 이동 객체의 개수 및 속도 그리고 k 값의 변화에 따른 모니터링 방법의 정확도

증가하면 최근접 쌍을 찾기 위한 탐색영역의 임계값이 증가하여 결과값의 변화율이 작아지기 때문에 정

확도는 떨어진다. 두 가지 일괄처리 방법에 비해 본 논문에서 제안하는 모니터링 방법의 성능이 더 나음을 알 수 있다.

6. 결론 및 향후연구

본 논문에서는 도로 네트워크 기반의 k 배타적 최근접 쌍을 구하기 위한 클러스터 기반 배타적 최근접 쌍 검색 방법과 이동 객체의 위치 변화에 따른 결과값의 갱신을 위한 모니터링 방법을 제안하였다. 그리고 2가지의 실제 도로 네트워크들을 기반으로 다양한 실험을 수행하여 본 논문에서 제안하는 방법론들의 성능을 평가하였다.

클러스터 기반 배타적 최근접 쌍 검색 방법은 이동 객체에 인접한 도로 위의 노드를 중심으로 검색 영역을 점차 확장하면서 최근접 쌍을 찾기 때문에 중복 검색 영역이 최소화되어 도로 네트워크 크기에 영향을 크게 받지 않았고 수행속도 또한 빨랐다. 최근접 쌍 결과에 대한 모니터링 방법은 이동 객체의 위치 변화에 대해 부분적인 검색 영역 확장을 통해 결과값을 갱신한다. 실험 결과 최근접 쌍 검색을 재수행하는 방법에 비해 정확도가 높음을 알 수 있었다. 본 논문의 기여는 다음과 같이 정리할 수 있다.

- 이동 객체 간의 배타적 최근접 쌍 검색 방법 및 결과 모니터링 방법을 제안하였다. 실제 응용 분야에서 이동 객체간의 배타적인 최근접 쌍을 검색하는 방법은 매우 유용하게 사용될 수 있다.
- 실제 도로 네트워크를 통한 다양한 실험을 통해 본 논문에서 제안하는 모니터링 방법의 정확성이 다른 일괄 처리 방법에 비해 높음을 보였다.

향후 연구로는 일방통행 길 또는 버스 전용차로와 같이 실 상황에 존재하는 여러 요소들을 고려하여 배타적 최근접 쌍 검색 방법의 기능을 확장하는 것이다.

사 사

이 논문은 부산대학교 자유과제 학술연구비(2년)에 의하여 연구되었음

참고문헌

- Xiaohui Yu, Ken Q. Pu and Nick Koudas, 2005, "Monitoring k-Nearest Neighbor Queries over Moving Objects", International Conference on Data Engineering, pp. 631-642.
- Kyriakos Mouratidis, Man Lung Yiu, Dimitris Papadias and Nikos Mamoulis, 2006, "Continuous Nearest Neighbor Monitoring in Road Networks", International Conference on Very Large Data Bases, pp. 43-54.
- Manli Zhu, Dik Lun Lee and Jun Zhang, 2006, "k-Closest Pair Query Monitoring over Moving Objects", International Conference on Mobile Data Management, pp. 14-22.
- Leong Hou U, Nikos Mamoulis and Man Lung Yiu, 2008, "Computation and Monitoring of Exclusive Closest Pair", IEEE Transactions on Knowledge and Data Engineering, vol 20, no 12, pp. 1641-1654.
- Kyriakos Mouratidis, Marios Hadjieleftheriou and Dimitris Papadias, 2005, "Conceptual Partitioning: An Efficient Method for Continuous Nearest Neighbor Monitoring", ACM SIGMOD International Conference on Management of Data, pp. 634-645.
- 김보련, 이상현, 이기준, 2005, "도로 네트워크 기반 이동 객체의 궤적 데이터 생성", 한국정보처리학회 추계 학술 발표 논문집(상), 12권 2호, pp. 75-78.
- 김준석, 이기준, 장병태, 유재준, 2007, "도로네트워크 기반의 3차원 객체를 위한 연속근원질의처리", 한국GIS학회지, 15권 2호, pp. 95-109.
- 김정수, 강혜영, 이기준, 2005, "P2P와 브로드캐스팅을 이용한 도로 네트워크 상에서 이동 노드의 위치정보관리 및 질의 처리", 한국GIS학회 추계학술대회, pp. 35-43.
- 지정희, 최보윤, 김상호, 류근호, 2004, "이동객체의 궤적에 대한 연속 최근접 질의 처리", 정보과학회논문지 제31권 5호, pp. 492-504.
- 홍동숙, 김동오, 이강준, 한기준, 2007, "도로 네트워크 환경에서 이동 객체 위치 예측을 위한 효율적인 인덱싱 기법", 한국공간정보시스템학회 제9권 2호, pp1-13.

접수일	(2009년 6월 17일)
최종수정일	(2009년 7월 21일)
게재확정일	(2009년 7월 28일)