

도로 네트워크를 위한 k -최근접 이웃 질의의 처리 방안

이상철 | 김상욱
한양대학교

요약

텔레매틱스에서 위치 정보 서비스를 효과적으로 제공하기 위해 이동 객체와 더불어 시설물과 같은 정적 객체에 대한 위치 정보를 효과적으로 관리하는 데이터베이스 기술들이 요구된다. 본 논문에서는 도로 네트워크 데이터베이스를 위한 인덱싱 및 질의 처리 기술 현황에 대하여 고찰한다. 텔레매틱스에서는 영역 질의, k -최근접 이웃 질의, 연속 k -최근접 이웃 질의, 공간 조인 질의 등이 발생하며, 이 중 k -최근접 이웃 질의가 빈발하게 발생한다. k -최근접 이웃 질의를 처리하기 위한 효과적인 방안으로 IER, INE, VN^3 , 근사 인덱싱 기법 등이 있다. 본 논문에서는 각 기법의 개념, 알고리즘, 장단점에 대하여 고찰한다.

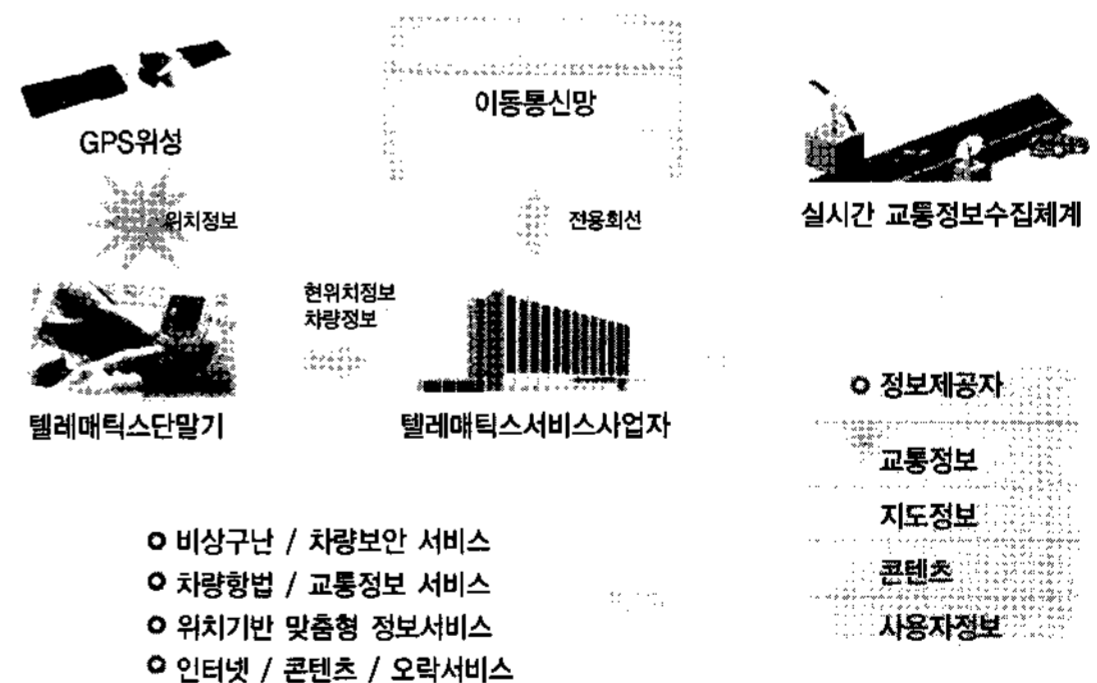
해당 자동차에서 가장 가까운 주유소와 주유에 관련된 정보 등을 검색하여 사용자에게 정보를 전송해 준다. 이러한 위치 정보 서비스를 효과적으로 제공하기 위해 이동 객체와 더불어 시설물과 같은 정적 객체에 대한 위치 정보를 효과적으로 관리하는 데이터베이스 기술들이 요구된다.

텔레매틱스를 위한 요구되는 데이터베이스 관련 기술로는 도로 네트워크 모델링, 인덱싱, 질의 처리, 이동 객체 궤적 분석 등이 있다. 본 논문에서는 도로 네트워크에서 사용되는 질의의 종류를 설명하고, 이 중 실제 응용에서 빈발하게 사용되는 k -최근접 이웃 질의 처리하는 다양한 기법들을 소개한다.

1. 서론

텔레매틱스는 무선 통신 기술, 위성 항법 장치 등을 이용하여 자동차, 선박 같은 이동 객체(moving object)에 위치 정보 서비스를 지원하는 응용 분야이다. 텔레매틱스가 제공하는 위치 정보 서비스로는 비상 구난, 차량 보안, 차량 항법, 교통 정보, 위치 기반 맞춤형 정보 서비스 등이 있다[1].

(그림 1)은 텔레매틱스를 위한 위치 정보의 흐름을 설명하는 그림이다. 예를 들어, 자동차의 연료가 부족한 경우 현재 자동차의 위치와 상태를 텔레매틱스 서비스 사업자에게 전송한다. 텔레매틱스 서비스 사업자는 해당 정보를 분석하여



(그림 1) 텔레매틱스를 위한 정보의 흐름도

본 논문의 구성은 다음과 같다. 제 2장에서는 도로 네트워크 데이터베이스를 위한 질의의 종류를 서술한다. 제 3장에서는 도로 네트워크 공간에서 k -최근접 이웃 질의를 위한 인

덱싱 및 질의 처리 대하여 서술한다. 마지막으로, 제 4장에서는 본 논문을 요약하고, 결론을 내린다.

II. 질의의 종류

본 장에서는 도로 네트워크 환경에서 대표적인 질의인 영역 질의, k -최근접 이웃 질의, 연속 k -최근접 이웃 질의, 공간 조인 질의를 설명하고, 그 예를 보이도록 한다.

(1) 영역 질의

영역 질의[2,3,4,5]는 질의 점으로부터 일정 거리 안에 존재하는 정적 객체들을 검색한다. 예를 들어, 자동차로부터 5km 이내에 있는 호텔들을 모두 검색하라는 질의가 이에 해당된다.

(2) k -최근접 이웃 질의

k -최근접 이웃 질의[6,7,8,9]는 질의 점으로부터 가까운 순으로 주어진 개수의 정적 객체들을 검색한다. 예를 들어, 자동차로부터 가까운 순으로 5개의 호텔들을 검색하라는 질의가 이에 해당된다.

(3) 연속 k -최근접 이웃 질의

연속 k -최근접 이웃 질의[10,11,12,13]는 질의점이 주어진 경로로 움직일 때, 각 위치에서 k -최근접 이웃 질의의 결과를 반환한다. 예를 들어, 자동차가 회사에서 집으로 가는 경로가 주어졌을 때, 경로를 따라가며 매순간 가장 가까운 3개의 주유소를 검색하라는 질의가 이에 해당된다. 이는 k -최근접 이웃 질의를 확장한 것으로써, 주어진 경로 내에서 k -최근접 이웃이 바뀌는 지점과 그 지점부터의 새로운 k -최근접 이웃을 저장함으로써 처리 될 수 있다.

(4) 공간 조인 질의

공간 조인 질의[14,15,16,17]는 서로 다른 두 종류의 정적 객체들의 집합을 대상으로 수행되며, 사이의 거리가 가장 가까운 쌍을 검색한다. 예를 들어, 거리가 가장 가까운 호텔과 음식점의 쌍을 검색하라는 질의가 이에 해당된다. 이러

한 질의는 데이터베이스에 저장되어 있는 호텔에 대한 위치 정보와 음식점에 대한 위치 정보를 이용하여 가장 가까운 쌍을 검색하게 된다.

텔레매틱스는 위에서 언급한 다양한 질의의 종류를 효과적으로 처리할 수 있어야 한다. 본 논문에서는 위에서 기술한 질의 중 실제 응용에서 빈발하게 사용되며, 활발하게 연구되고 있는 k -최근접 이웃 질의에 대해 초점을 맞추어 논의한다.

III. k -최근접 이웃 질의의 처리

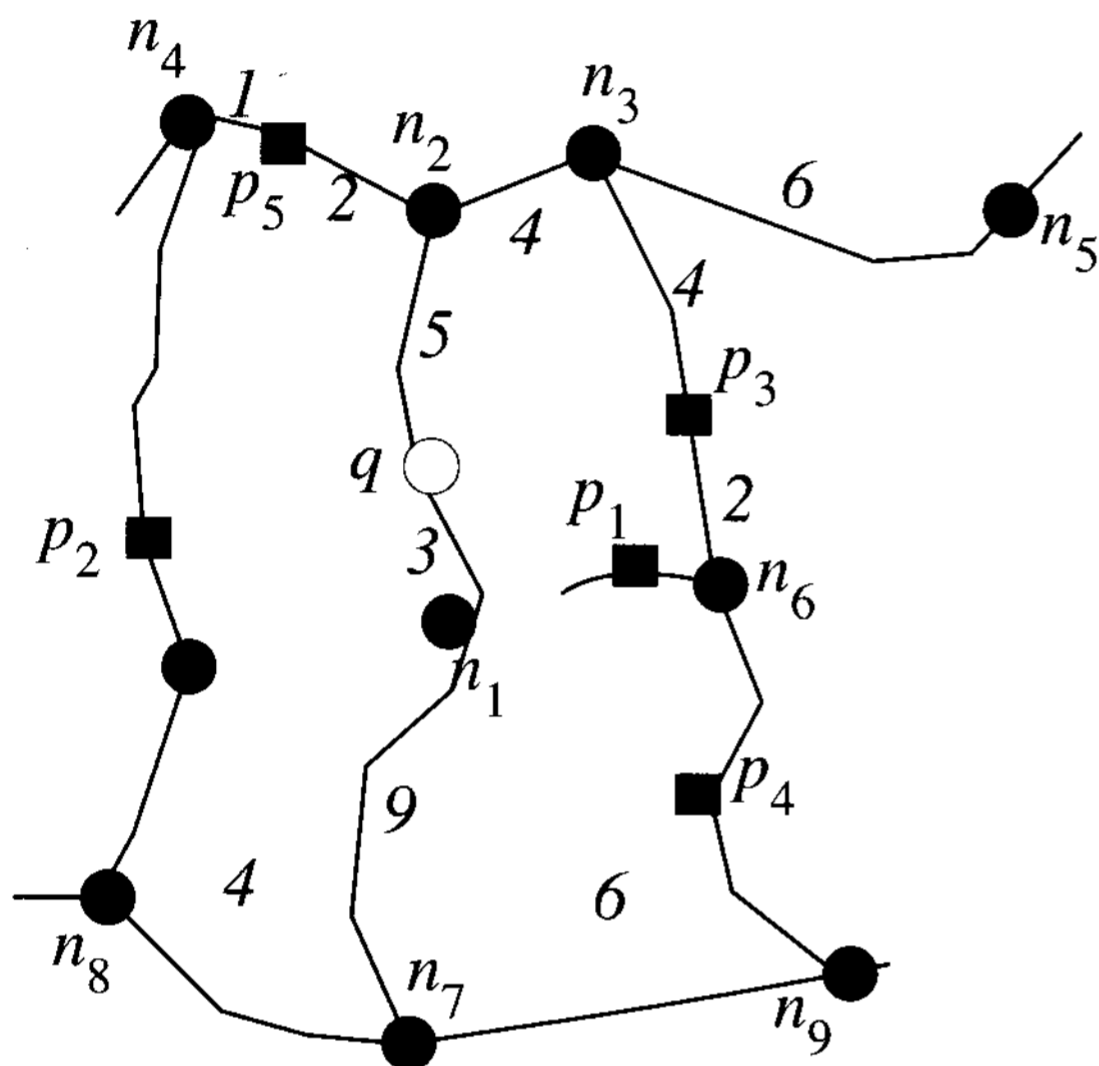
도로 네트워크에서 질의 처리에 대한 연구는 이동 객체의 현재 위치에서 가장 가까운 k 개의 정적 객체들을 검색하는 k -최근접 이웃 질의에 초점을 맞추고 있다. k -최근접 이웃 질의를 처리하기 위한 대표적인 방식들은 INE[18], IER[18], VN3[19], 네트워크 거리 기반 근사 인덱싱(approximate indexing)[20] 등이 있다.

3.1 INE

INE(incremental network expansion)[18]는 질의 점이 주어졌을 때, 우선 질의 점이 포함된 도로 세그먼트를 찾는다. 그 후, 해당 도로 세그먼트로부터 시작하여 인접 도로 세그먼트를 하나씩 확장해 가며, 해당 도로 세그먼트에 정적 객체가 있는지의 여부를 정적 객체가 저장되어 있는 R-트리[21, 22]를 이용하여 검색한다. 위의 과정을 사용자가 요구한 k 개의 정적 객체를 찾을 때까지 반복적으로 도로 세그먼트를 확장한다.

(그림 2)와 같이 도로 네트워크와 정적 객체가 존재할 때, 질의 점 q 가 주어졌다고 가정하자. (그림 2)에서 사각형 점은 정적 객체를 의미하고, 원형 점은 도로 세그먼트의 양 끝 노드를 나타낸다. INE를 통하여 1-최근접 이웃 질의를 처리하고자 할 때 먼저 질의 점을 포함하고 있는 도로 세그먼트를 검색한다. 이미 도로 세그먼트로 구성된 2차원의 R-트리가 구축되어 있기 때문에 이를 통해 쉽게 질의 점을 포함하고 있는 도로 세그먼트를 검색할 수 있다. 검색된 도로 세그먼트의 양 끝 노드를 가까운 거리에 따라 우선순위 큐에 삽

입 한다. 또한, 도로 세그먼트에 정적 객체 유무를 판단하기 위하여 검색된 도로 세그먼트의 위치를 정적 객체가 저장된 R -트리에 질의한다. (그림 2)의 예에서는 정적 객체가 발견되지 않았기 때문에 가장 가까운 노드를 우선순위 큐에서 팝(pop)하여 해당 노드와 연결된 다른 도로 세그먼트를 검색한다.



(그림 2) INE를 통한 k -최근접 이웃 질의 처리

INE에서는 도로 세그먼트의 한 노드와 다른 도로 세그먼트와의 연결 관계를 링크드 리스트로 관리한다. 이에 따라 노드 n_1 에서 도로 세그먼트 n_1n_7 로 쉽게 확장할 수 있다. 확장한 도로 세그먼트의 노드 n_7 을 우선순위 큐에 삽입한 후 확장된 도로 세그먼트에 정적 객체가 있는지를 정적 객체가 저장된 R -트리를 이용하여 검색한다. 이와 같은 방법으로 도로 세그먼트 n_2n_4 와 n_2n_3 로 확장하고 각 도로 세그먼트에 정적 객체가 있는지의 여부를 정적 객체가 저장된 R -트리를 통하여 검색을 수행한다.

이때 도로 세그먼트 n_2n_4 에 정적 객체 p_5 가 발견되었기 때문에 질의 점부터 p_5 까지의 거리를 계산한다. 이 거리가 질의 점부터 다음 팝 될 노드까지의 거리보다 클 경우 p_5 가 가장 가까운 정적 객체이기 때문에 결과를 반환하고 질의를 종료한다. 그렇지 않을 경우에는 p_5 보다 가까운 정적 객체가 존재할 수 있기 때문에 도로 세그먼트를 확장하여 질의를 계속 수행한다. (그림 2)에서 질의 점에서 p_5 까지 거리는 7

이며, 다음 팝 할 노드 n_4 의 거리는 8임으로 p_5 를 결과로 반환하고 질의 처리를 종료한다.

<표 1>은 INE를 통한 k -최근접 이웃 질의 처리 과정을 의사코드로 나타낸 것이다. 이 방식은 저장 공간이 작다는 장점을 가지고 있지만, 도로 세그먼트를 확장할 때마다 도로 세그먼트에 대한 정보를 디스크로부터 액세스해야 한다. 또한, 정적 객체 유무에 대한 결과를 R -트리를 통하여 매번 검색해야 하므로 질의 성능이 떨어진다.

<표 1> INE 알고리즘

Algorithm INE

- (1) 질의 점이 위치한 도로 세그먼트를 찾는다.
- (2) 단계 1에서 검색된 도로 세그먼트 상에 정적 객체가 있는지를 검색한다.
- (3) 정적 객체가 발견 되었으면 질의 점과 가까운 순으로 결과 집합을 유지하며, 가장 먼 거리를 $d_{N_{max}}$ 로 유지한다. 만약 k 번째 정적 객체가 발견되지 않았다면 $d_{N_{max}}$ 는 무한수 ∞ 를 갖는다.
- (4) 단계 1에서 검색된 도로 세그먼트의 양 끝 노드를 질의와 거리가 가까운 순으로 우선순위 큐에 삽입한다.
- (5) 우선순위 큐에서 다음 노드를 팝(pop)하여 도로 세그먼트를 확장한다. 만약 $d_{N_{max}}$ 가 확장될 노드와 질의 점까지의 거리보다 작다면 현재까지의 결과 집합을 반환하고 종료한다.
- (6) 확장된 도로 세그먼트에 정적 객체가 있다면 정적 객체와 질의 점까지의 거리와 $d_{N_{max}}$ 와 비교하여 결과 집합에 삽입 여부를 판단한다.
- (7) 만약, 발견된 정적 객체가 삽입되어서 질의 점에서 k 번째로 가까운 정적 객체가 바뀌었다면 $d_{N_{max}}$ 을 k 번째 정적 객체까지의 거리로 변경한다.
- (8) 확장된 노드는 우선순위 큐에 삽입한다.
- (9) 단계 5- 단계 8을 반복한다.

3.2 IER

IER(incremental Euclidean restriction)[18]은 두 점간의 네트워크 거리는 유클리드 거리보다 항상 크거나 같다는 특성을 이용한다. 우선, 정적 객체가 저장되어 있는 R -트리를 이용하여 유클리드 거리를 기준으로 질의 점에서 가장 가까운

k 개의 후보 정적 객체들을 검색한다. 그 후, 후보 정적 객체들에서 질의 점까지의 실제 네트워크 거리를 계산하여, 그 중 가장 큰 값을 반지름으로 하는 영역 질의를 수행한다. 현 단계에서 검색된 정적 객체에서 이전 단계의 후보 정적 객체들을 제외하고, 유클리드 거리를 기준으로 가장 가까운 정적 객체를 찾아 질의 점과의 네트워크 거리를 계산한다. 이 거리가 후보 정적 객체들 중 질의 점과의 네트워크 거리가 가장 큰 정적 객체보다 가까우면, k 번째 정적 객체는 후보에서 탈락시키고 새로 검색된 정적 객체를 k 개의 후보 정적 객체에 넣는다. 위의 과정을 계속 반복하여 질의 점에서 가장 가까운 k 개의 정적 객체를 검색한다.

(그림 3)은 IER를 통하여 1-최근접 이웃 질의를 처리하는 예를 나타낸다. (그림 3)에서 사각형 점은 정적 객체를 의미하며, 원형 점은 질의 점을 의미한다. IER는 질의 점 q 가 주어졌을 때 먼저 질의 점과 유클리드 거리상으로 가장 가까운 정적 객체 P_{E1} 을 찾는다. INE와 동일하게 정적 객체에 대한 R -트리를 유지하고 있기 때문에 쉽게 검색이 가능하다. 이 후에 질의 점부터 정적 객체 P_{E1} 까지의 실제 네트워크 거리를 계산한다. 해당 정적 객체 P_{E1} 의 네트워크 거리를 $d_{E_{max}}$ 로 정의하고 유클리드 거리를 d_E 로 정의한다. 질의 점부터 d_E 와 $d_{E_{max}}$ 사이에 정적 객체를 영역 질의를 통하여 검색한다. 검색된 정적 객체들의 네트워크 거리를 모두 계산하여 네트워크 거리가 $d_{E_{max}}$ 보다 작은지 여부를 확인한다. (그림 3)의 경우 질의 점부터 정적 객체 P_{E2} 의 네트워크 거리가 $d_{E_{max}}$ 보다 작기 때문에 결과 집합에서 P_{E1} 을 제외하고 P_{E2} 를 삽입한 후 $d_{E_{max}}$ 와 d_E 값을 정적 객체 P_{E2} 에 맞추어 변경한다. P_{E3} 의 경우 변경된 $d_{E_{max}}$ 보다 유클리드 거리가 크기

때문에 제외하고 객체 P_{E2} 를 결과로 반환하고 질의 처리를 종료한다.

〈표 2〉는 IER를 통한 k -최근접 이웃 질의 처리 과정을 의사 코드로 나타낸 것이다. 이 방식은 INE와 비슷한 저장 공간을 필요로 하지만 여러 번의 네트워크 거리를 계산하는 시행 착오를 겪기 때문에 질의 처리 성능이 크게 떨어진다[18].

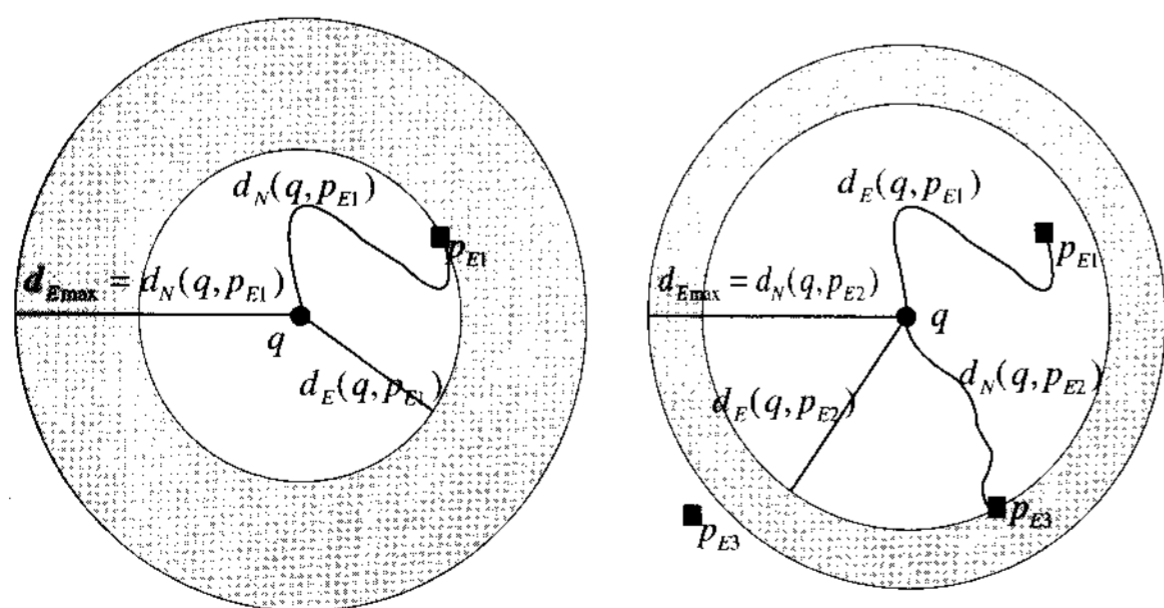
3.3 VN³

VN³(Voronoi network nearest neighbor)[19]는 하나의 정적 객체와 인접한 다른 정적 객체간의 거리가 같은 지점을 경계로 셀을 나누어 두는 방식이다. 셀의 중심에는 한 개의 정적 객체만이 존재하게 되고, 셀 안의 어느 지역에서나 가장 가까운 정적 객체는 셀 중심에 있는 정적 객체이다. 셀 내적으로는 노드와 정적 객체, 경계점과 정적 객체, 그리고 셀 안의 모든 경계점간 거리가 미리 계산되어 있다. 또한, 이외에도 인접 셀에 대한 정보가 저장되어 있다. k -최근접 질의 처리 시, k 가 1일 때는 질의 점이 있는 셀 안의 정적 객체가 가장 가까운 정적 객체가 된다. k 가 1이상일 때에는 해당 셀을 찾고 인접 셀들을 후보로 둔다. 질의 점과 질의 점이 포함된 셀의 경계까지의 거리를 계산하면, 후보의 셀 중 어떤 셀 안의 정적 객체가 두 번째로 가까운 정적 객체인지 알게 된다. 이렇게 두 번째 셀을 발견하게 되면, 두 번째 셀의 인접 셀 역시 후보가 된다. k 개의 정적 객체가 다 찾아 질 때까지 위의 질의 처리 과정을 반복한다.

〈표 2〉 IER 알고리즘

Algorithm IER

- (1) 질의 점으로부터 유클리드 거리상으로 가장 가까운 k 개의 정적 객체를 찾으며, 유클리드 거리가 가장 먼 객체까지의 유클리드 거리를 d_E 로 설정한다.
- (2) 단계 1에서 검색된 k 개의 정적 객체까지의 네트워크 거리를 계산하여 질의 점과 가까운 순으로 유지한다.
- (3) $d_{E_{max}}$ 를 k 번째 정적 객체의 네트워크 거리로 설정한다.
- (4) 질의 점으로부터 d_E 와 $d_{E_{max}}$ 사이의 정적 객체를 검색한다.
- (5) 질의 점으로부터 유클리드 거리 d_E 가 가까운 순으로 정적 객체의 네트워크 거리를 계산한다.
- (6) 네트워크 거리가 $d_{E_{max}}$ 와 비교하여 작으면 결과 셋에

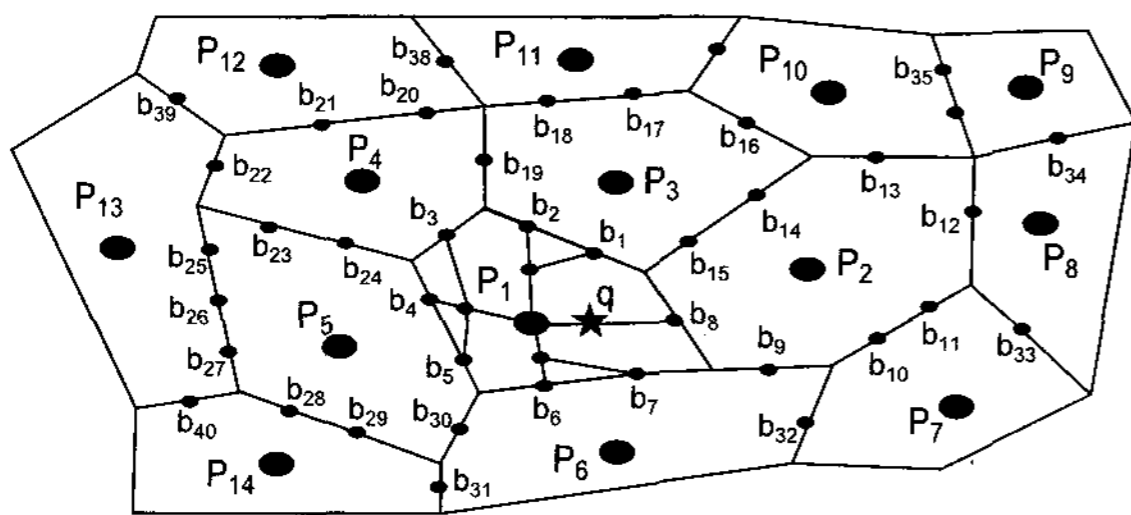


(그림 3) IER을 통한 k -최근접 이웃 질의 처리의 예

삼입하고 k 번째 정적 객체의 네트워크 거리를 $d_{E_{max}}$ 로 설정한다.

- (7) 검색된 정적 객체의 유클리드 거리 d_E 가 $d_{E_{max}}$ 보다 클 때 까지 단계 5와 단계 6을 반복한다.

(그림 4)는 VN^3 를 이용하여 k -최근접 이웃 질의 처리의 예를 나타낸다. 우선, 도로 네트워크를 정적 객체 P_1 부터 P_{14} 까지 서로의 거리를 기준으로 (그림 4)와 같은 셀로 구분한다. 이때, 인접한 두 정적 객체간의 거리가 같은 지점을 경계점(border point)라고 하며 b_i 로 표기한다. 이렇게 셀을 구분한 이후 셀의 중심에 있는 정적 객체와 경계점까지의 거리, 정적 객체와 한 셀 안의 노드까지의 거리, 그리고 노드와 경계점까지의 거리를 계산하여 저장해 둔다. 또한, 각 셀은 해당 셀에 인접한 셀에 대한 정보를 유지하고, 도로 세그먼트와 해당 도로 세그먼트가 속한 셀의 정보를 R -트리로 관리한다. 질의 점 q 와 검색할 개수 k 가 주어졌을 때 우선 질의 점을 포함하는 도로 세그먼트를 R -트리를 통하여 검색하고, 검색된 도로 세그먼트가 속한 셀을 알아낸다. 그리고 검색된 셀 안의 정적 객체를 1-최근접 이웃 질의의 결과로 반환하고, 인접 셀에 속한 정적 객체를 후보 집합으로 둔다. 질의 점 q 를 포함하는 도로 세그먼트의 양 끝 노드와 해당 셀의 경계선까지의 거리를 이용하여 다음으로 가까운 정적 객체를 찾아낸다. 그리고 해당 정적 객체가 있는 셀의 인접한 정적 객체를 후보로 둔다. k 개의 정적 객체를 모두 찾으면 그 결과를 반환하고 질의 처리를 종료한다.



(그림 4) VN^3 를 통한 k -최근접 이웃 질의 처리의 예

<표 3>과 <표 4>는 VN^3 를 생성하는 과정과 k -최근접 이웃 질의 처리 과정을 의사 코드로 나타낸 것이다. 이 방식은 INE, IER과 비교하여 질의 처리 성능이 매우 뛰어나다. 그러

나 VN^3 를 구축하기 위하여 미리 계산하고 저장해야 할 정보가 많기 때문에 저장 공간의 오버헤드가 매우 심하다는 단점이 있다.

<표 3> VN^3 생성 알고리즘

Algorithm VN^3 Build

- (1) 모든 정적 객체와 모든 노드간의 네트워크 거리를 계산한다.
- (2) 모든 노드에서 가장 가까운 정적 객체가 같은 노드끼리 하나의 셀로 묶는다.
- (3) 도로 세그먼트의 양 끝 노드에서 가장 가까운 정적 객체가 다른 도로 세그먼트가 다르다면 해당 도로 세그먼트에서 경계점을 구한다.
- (4) 단계 3에서 각 셀의 인접 셀을 구하여 관리한다.
- (5) 각 셀에서 경계점과 정적 객체까지 거리를 계산하여 저장한다.
- (6) 각 셀에서 경계점과 셀 안의 모든 노드까지 거리를 계산하여 저장한다.
- (7) 각 셀 안의 경계점간의 거리를 계산하여 저장한다.
- (8) 도로 세그먼트 R -트리를 구축한다.

<표 4> VN^3 질의 처리 알고리즘

Algorithm VN^3 Query Processing

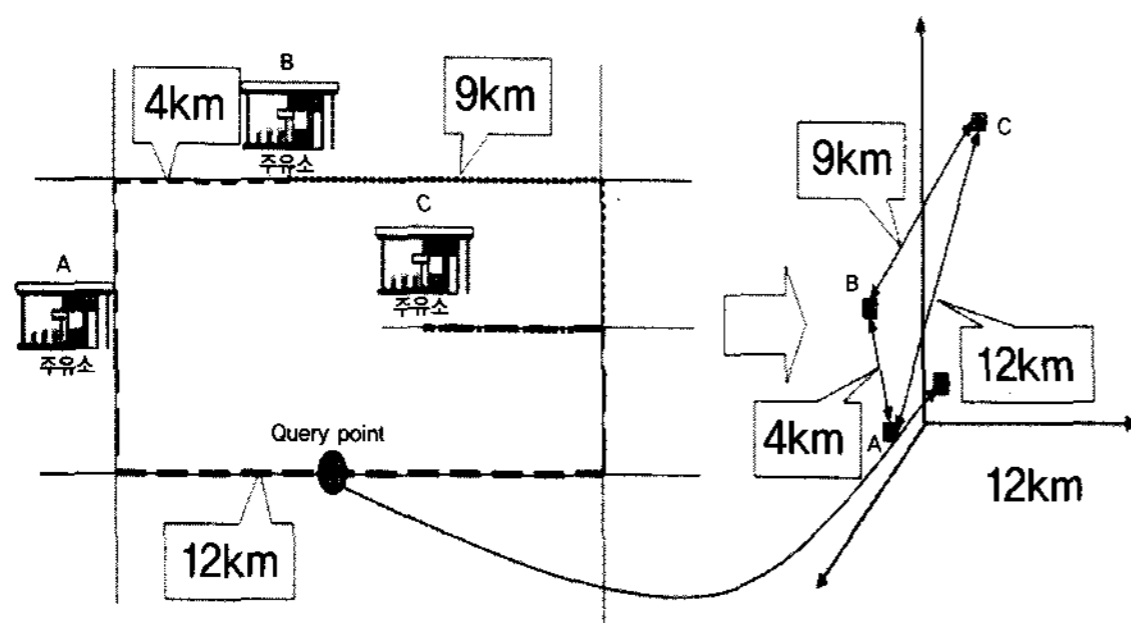
- (1) 질의 점이 속한 도로 세그먼트를 검색한다.
- (2) 질의 점이 속한 셀을 단계 1에서 찾아낸 도로 세그먼트 정보를 이용하여 구한다.
- (3) 해당 셀에 있는 정적 객체를 결과 집합에 넣는다.
- (4) k 가 1이라면 결과 집합을 반환하고 종료한다.
- (5) 결과 집합에 있는 정적 객체가 속한 셀의 인접 셀을 후보 집합으로 유지한다.
- (6) 질의 점으로부터 각 후보 셀 안의 정적 객체까지의 거리를 생성 시점에 저장된 정보를 이용하여 계산한다.
- (7) 그중 가장 가까운 정적 객체를 결과 집합에 넣는다.
- (8) 결과 집합 안에 정적 객체의 개수가 k 개보다 작으면 단계 5 - 단계 7을 반복한다.

3.4 네트워크 거리 기반 근사 인덱싱

네트워크 거리 기반 근사 인덱싱(approximate indexing)

[20]은 각 정적 객체를 FastMap 알고리즘[23]을 통하여 m -차원 유클리드 공간 상의 한 점으로 매핑한다. FastMap은 모든 객체 간 거리를 실제 거리와 근접하게 유지하면서 각 객체를 m -차원 유클리드 공간 상의 한 점으로 표현하는 기법 중 하나이다. m -차원 유클리드 공간으로 매핑된 객체 점들을 인덱스로 구축한다. k -최근접 이웃 질의를 처리하기 위해서 우선 질의 점을 FastMap 알고리즘을 이용하여 정적 객체가 매핑된 공간 상의 한 점으로 매핑한다. 이때, 질의 점은 모든 정적 객체 간의 실제 네트워크 거리가 유지되도록 매핑되기 때문에 한 번의 근사 인덱스 검색을 통하여 질의 처리가 가능하다.

(그림 5)는 네트워크 공간 내의 정적 객체 및 질의 점을 m -차원 유클리드 공간 상으로 매핑하여 질의 처리하는 과정을 나타낸다. (그림 5)의 왼쪽에서 도로 네트워크가 주어졌을 때, 원형 점은 질의 점을 의미하고 A, B, C는 정적 객체들을 의미한다. 우선, 모든 정적 객체 쌍 간의 네트워크 거리를 이용하여 주유소들의 위치를 m -차원 유클리드 공간 상의 절대 좌표로 매핑한다. 매핑 후, m -차원 유클리드 공간 상의 임의의 두 정적 객체 간 유클리드 거리는 원래의 네트워크 거리를 근사하게 된다. m -차원의 절대 좌표로 매핑된 정적 객체를 R -트리를 이용하여 인덱싱한다. 질의 처리를 위해서 질의 점 역시 위와 동일한 방법으로 m -차원 유클리드 공간 상의 절대 좌표로 매핑시킨다. (그림 5)의 오른쪽에 A, B, C 정적 객체와 질의 점이 m -차원 상의 절대 좌표로 매핑된 결과를 보이고 있다. m -차원 상의 절대 좌표로 매핑된 질의 점과 정적 객체의 절대 좌표로 구축된 인덱스를 이용하여 k -최근접 이웃 질의를 처리를 수행한다.



(그림 5) 근사 인덱싱을 통한 k -최근접 이웃 질의 처리의 예

(표 5)와 (표 6)은 근사 인덱싱을 생성하는 과정과 k -최근접 이웃 질의 처리 과정을 의사코드로 나타낸 것이다. 근사 인덱싱 기법은 INE, IER 기법에 비해 효과적인 질의 처리가 가능하며, VN^3 에 비해 적은 공간 오버헤드를 갖는다. 그러나 m -차원으로 정적 객체와 질의 점을 매핑하는 과정에서 오차가 생길 수 있다.

〈표 5〉 근사 인덱싱 알고리즘

Algorithm Approximate Indexing

- (1) 모든 정적 객체 쌍의 네트워크 거리를 구한다.
- (2) FastMap을 이용하여 각 정적 객체를 m -차원 상의 한 점으로 매핑한다.
- (3) R -트리를 이용하여 m -차원 유클리드 공간 상의 모든 정적 객체들을 인덱싱한다.
- (4) 각 노드에서 모든 피봇까지의 평균 네트워크 거리를 구하여 저장한다.

〈표 6〉 근사 인덱싱을 이용한 질의 처리 알고리즘

Algorithm Query Processing

- (1) 질의 점이 위치한 도로 세그먼트를 찾는다.
- (2) 질의 점에서 피봇 객체들까지의 평균 네트워크 거리를 구한다.
- (3) FastMap을 이용하여 질의 점을 m -차원 유클리드 공간 상의 절대 좌표 점으로 매핑시킨다.
- (4) 정적 객체를 위한 m -차원 인덱스를 통하여 k -최근접 이웃 질의를 처리한다.

IV. 결 론

텔레매틱스는 이동 객체 및 정적 객체의 데이터를 활용하여 사용자에게 유용한 정보를 제공하는 서비스이다. 텔레매틱스 사용자에게 다양한 정보를 제공하기 위해서는 이동 객체와 정적 객체에 대한 위치 정보를 이용한 다양한 질의 처리를 지원해야 한다. 본 논문에서는 도로 네트워크 환경에서 빈발하게 이용되고 있는 k -최근접 이웃 질의에 대한 연구

동향에 관하여 논의 하였다.

효과적으로 k -최근접 이웃 질의를 처리하기 위하여 INE, IER, VN^3 , 그리고 근사 인덱싱 등의 기법이 이용되고 있다. INE와 IER 기법은 비교적 적은 공간 오버헤드를 가지는 반면 질의 처리 성능이 떨어진다. VN^3 는 빠른 질의 처리 성능을 보장하지만, 저장 공간 오버헤드가 심하다. 근사 인덱싱 기법의 경우 VN^3 보다는 적은 공간 오버헤드를 요구하면서 빠른 질의 처리가 가능하다. 그러나 정확한 질의 결과를 보장하기 위한 후처리가 필요하다. 한양대학교 데이터마이닝 및 지식공학 연구실에서는 정확한 질의 결과 보장을 위한 후처리 방안에 관하여 현재 연구를 진행 중에 있다

본 논문에서 살펴본 바와 같이 현재 도로 네트워크 공간상의 도로 네트워크 데이터베이스 기술에 대한 연구는 활발히 진행되고 있지만, 그 수준은 아직 높지 않은 상태이다. 산학연이 더욱 분발하여 훌륭한 연구 결과들을 도출함으로써 실제 환경에서 유용한 텔레매틱스 서비스를 누릴 수 있기를 기대한다.

감사의 글

본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT 연구센터 지원사업의 연구결과로 수행되었음(IITA-2008-C1090-0801-0040).



- [1] S. Duri et al., "Data Protection and Data Sharing in Telematics," *Mobile Networks and Applications*, MONET, Vol. 9, No. 6, pp. 693-701, 2004.
- [2] Y. Theodoridis and D. Papadias, "Range Queries Involving Spatial Relations: A Performance Analysis," In *Proc. Int'l Conf. on Spatial Information Theory*, COSIT, pp. 537-551, 1995.
- [3] G. Proietti and C. Faloutsos, "I/O Complexity for Range Queries on Region Data Stored Using an R-tree," In *Proc. Int'l Conf. on Data Engineering*, ICDE, pp. 628-635, 1999.
- [4] B. Pagel et al., "Towards an Analysis of Range Query Performance in Spatial Data Structures," In *Proc. Int'l Symp. on Principles of Database Systems*, PODS, pp. 214-221, 1993.
- [5] D. Papadias et al., "Multidimensional Range Query Processing with Spatial Relations," *Journal of Geographical Systems*, Vol. 4, No. 4, pp. 343-365, 1997
- [6] C. Jensen et al., "Nearest Neighbor Queries in Road Networks," In *Proc. Int'l Symp. on Advances in geographic information systems*, ACM GIS, pp. 1-8, 2003.
- [7] Z. Song and N. Roussopoulos, "K-Nearest Neighbor Search for Moving Query Point," In *Proc. Int'l Symp. on Advances in Spatial and Temporal Databases*, SSTD, pp.79-96, 2001.
- [8] S. Chaudhuri and L. Gravona, "Evaluating Top-k Selection Queries," In *Proc. Int'l Conf. on Very Large Database*, VLDB, pp. 397-410, 1999.
- [9] N. Roussopoulos et al., "Nearest neighbor queries," In *Proc. Int'l Conf. on Management of Data*, SIGMOD, pp. 71-79, 1995.
- [10] Y. Tao et al., "Continuous Nearest Neighbor Search," In *Proc. Int'l Conf. on Very Large Data Bases*, VLDB, pp. 287-298, 2002.
- [11] G. Iwerks et al., "Continuous k-Nearest Neighbor Queries for Continuously Moving Points with Updates," In *Proc. Int'l Conf. on Very Large Data Bases*, VLDB, pp. 512-523, 2003.
- [12] Y. Li et al., "Continuous K Nearest Neighbor Queries in Spatial Network Databases," *GeoInformatica*, Vol. 9, No. 4, pp. 321-341, 2005.
- [13] M. Kolahdouzan and C. Shahabi, "Alternative Solutions for Continuous K Nearest Neighbor Queries in Spatial Network Databases," *GeoInformatica*, Vol. 9, No. 4, pp. 321-341, 2005.
- [14] T. Brinkhoff et al., "Efficient Processing of Spatial Joins Using R-trees," In *Proc. ACM Int'l Conf. on Management of Data*, ACM SIGMOD, pp. 237-246, 1993.

- [15] C. Faloutsos et al., "Spatial Join Selectivity Using Power Laws," In *Proc. ACM Int'l. Conf. on Management of Data*, ACM SIGMOD, pp. 177-188, 2000.
- [16] N. Mamoulis and D. Papadias, "Slot Index Spatial Join," *IEEE Trans. on Knowledge and Data Engineering*, TKDE, Vol. 15, No. 1, pp. 211-231, 2003.
- [17] N. Mamaulis and D. Papadias, "Multiway Spatial Joins," *ACM Trans. on Database Systems*, TODS, Vol. 26, No. 4, pp. 424-475, 2001.
- [18] D. Papadias et al., "Query Processing in Spatial Network Databases," In *Proc. Int'l. Conf. on Very Large Data Bases*, VLDB, pp. 802-813, 2003.
- [19] M. Kolahdouzan and C. Shahabi, "Voronoi-Based K-Nearest Neighbor Search for Spatial Network Databases," In *Proc. Int'l. Conf. on Very Large Data Bases*, VLDB, pp. 840-851, 2004.
- [20] 이 상철, 김 상욱, "도로 네트워크 데이터베이스를 위한 근사 인덱싱," *한국정보처리학회 춘계학술발표대회 논문집*, Vol. 14, No. 1, pp. 61-62, 2007년 5월.
- [21] A. Guttman, "R-Trees: A Dynamic Index Structure for Spatial Searching," In *Proc. ACM Int'l. Conf. on Management of Data*, ACM SIGMOD, pp. 47-57, 1984.
- [22] N. Beckmann et al., "The R*-Tree: An Efficient and Robust Access Method for Points and Rectangles," In *Proc. ACM Int'l. Conf. on Management of Data*, ACM SIGMOD, pp. 322-331, 1990.
- [23] C. Faloutsos and K. Lin, "FastMap: A Fast Algorithm for Indexing, Data-Mining and Visualization of Traditional and Multimedia Datasets," In *Proc. ACM Int'l. Conf. on Management of Data*, ACM SIGMOD, pp. 163-174, 1995.

약 력



이 상 철

2005년 한양대학교 학사
 2007년 한양대학교 석사
 2007년 ~ 현재 한양대학교 박사과정
 관심분야: 데이터베이스 시스템, 데이터 마이닝, 공간 데이터베이스/GIS, 이동 객체 데이터베이스/텔레매틱스, 웹 검색 엔진



김 상 욱

1989년 서울대학교 학사
 1991년 한국과학기술원 석사
 1994년 한국과학기술원 박사
 1991년 미국 Stanford University, Computer Science Department 방문 연구원
 1995년 KAIST 정보전자연구소 전문연구원
 2000년 미국 IBM T.J. Watson Research Center Post-Doc.
 2000년 강원대학교 부교수
 2003년 ~ 현재 한양대학교 교수

관심분야: 데이터베이스 시스템, 저장 시스템, 트랜잭션 관리, 데이터 마이닝, 멀티미디어 정보 검색, 공간 데이터베이스/GIS, 주기억장치 데이터베이스, 이동 객체 데이터베이스/텔레매틱스, 사회 연결망 분석, 웹 데이터 분석

