

# 모바일 객체의 방향성을 고려한 최근접 질의 처리† Nearest Neighbor Query Processing using the Direction of Mobile Object

이응재\*, 정영진\*, 최현미\*, 류근호\*, 이성호\*\*

Eungjae Lee, YoungJin Jung, HyonMi Choi, KeunHo Ryu, SeongHo Lee

**요약** 최근접 질의 (NN: Nearest Neighbor Query)는 질의 요청자와 가장 가까운 곳에 위치한 대상 객체를 검색하기 위한 질의로서, 모바일 환경에서 빈번하게 사용되는 질의 유형이다. 이 논문에서는 모바일 환경에서 방향 성분을 가지며 연속적으로 이동하는 질의 요청자가 요구하는 최근접 대상 객체를 검색하기 위한 질의 처리 방법을 제안한다. 제안된 방법은 모바일 환경에서 특정 방향 성분을 갖고 위치를 이동하는 질의 요청자의 방향 속성을 반영하여 최근접 객체를 검색할 수 있도록 유클리디안 거리 정보뿐만 아니라 사용자의 진행 방향을 고려하여 최근접 대상 객체를 검색한다. 제안된 방법은 모바일 환경에서 최근접 객체의 검색 기능을 요구하는 교통 정보 시스템, 관광정보 시스템, 위치 기반 추천 시스템과 같은 응용 분야에 적용할 수 있다.

**ABSTRACT** Nearest neighbor query retrieves nearest located target objects, and is very frequently used in mobile environment. In this paper, we propose a novel nearest neighbor query processing technique that is able to retrieve nearest located target object from the user who is continuously moving with a direction. The proposed method retrieves objects using the direction property of moving object as well as euclidean distance to target object.

The proposed method is applicable to traffic information system, travel information system, and location-based recommendation system which require retrieving nearest located object.

**주요어** : 최근접 질의, 이동 객체 데이터베이스 추천시스템, 위치기반 서비스

**Key word** : Nearest Neighbor Query, Moving Object Database, Recommendation System, LBS(Location-Based Services)

## 1. 서론

최근 무선 이동 통신 환경의 발달로 인하여 모바일 환경에서 사용자의 위치를 실시간으로 추적할 수 있는 GPS(Global Positioning System) 사용이 보편화되었다. 이러한 위치 측위 기술의 발달로 인하여 사용자의 위치에 기반을 둔 서비스 개발이 활발히 진행되고 있다. 위치기반 서비스 (LBS: Location-Based Services)는 이동하는 사용자의 위치를 파악하여 위치와 관련된 다양한 정보를 제공하고자 하는 서비스로서, 물류 차량 관리, 교통 통제 시스템, 차량의 위치를 관제 센터에서 실시간으로 모니터링 하는 차량 위치 추적 시스템[1][2][3] 등이 대표적인 응용 사례이다.

위치기반서비스를 효과적으로 지원하기 위해서는 실시간으로 제공되는 사용자의 위치를 관리하고, 그 정보를 제공하기 위한 기술이 필수적이다. 최근접 질의는 질의 요청자와 가장 가까이 위치한 객체를 검색하기 위한 질의로서 모바일 환경에서 빈번하게 사용되는 질의 중에 하나이다. 모바일 환경에서는 질의 요청자 또는 대상 객체가 시간의 흐름에 따라 그 위치가 연속적으로 이동한다는 특성을 갖기 때문에 연속적인 질의 처리 요청, 이동 방향에 따른 질의 처리 결과의 변화 등과 같은 새로운 형태의 질의 처리 요구가 발생된다. 하지만 기존의 공간 데이터베이스의 경우, 최근접 질의는 단순히 질의 객체와 대상 객체 사이의 거리를 계산하여 가장 가까운 대상 객체를 질의 결과로서 반환

† 이 연구는 ETRI 텔레매틱스 연구단과 과학기술부·한국과학재단 지정 청주대학교 정보통신연구센터의 지원에 의한 것입니다.

\* 충북대학교 전기전자컴퓨터공학부

{eungjae, yijeong, hmchoi, khryu}@dblab.chungbuk.ac.kr}

\*\* 한국전자통신연구원(ETRI) 텔레매틱스연구단

sholee@etri.re.kr

하도록 처리하기 때문에 모바일 환경의 이러한 요구 사항들을 효과적으로 처리하기 어렵다.

모바일 환경에서는 질의 객체 또는 대상 객체, 또는 두 객체 모두가 연속적으로 이동하기 때문에 시간이 지남에 따라 연속적인 질의 요청이 발생하며, 기존의 질의 처리 방법으로는 질의 처리 시스템에 매우 큰 부하를 주기 때문에 실시간 처리가 어려운 문제가 발생한다. 또한 객체의 이동 특성에 의하여 질의 요청 시간에서의 사용자 위치와 질의 결과 획득 시간에서의 위치가 달라지기 때문에 질의 결과 획득 시점에서의 사용자 환경에 적합하지 않은 질의 결과를 반환하게 된다. 예를 들어 콜택시 회사에서 고객으로부터 호출 서비스 요청을 받았을 경우, 고객과 가장 가까운 위치에 있는 택시가 고객이 있는 위치로 이동할 수 있도록 하여야 한다. 회사의 처리 시스템은 기존의 공간데이터베이스에서의 최근접 질의 처리 방법으로 고객과 가장 가까운 위치의 택시를 검색하여 명령을 내리게 된다. 하지만 검색된 택시의 이동 방향, 도로 정보 등에 의하여 검색된 차량보다 다른 차량이 먼저 고객에게 도착할 수 있는 경우가 빈번하게 발생한다. 따라서 기존의 최근접 처리 기법에 의해 검색된 차량은 고객으로의 이동 비용이 훨씬 많이 소요되는 문제가 발생하게 된다.

이 논문에서는 모바일 환경의 추천 시스템을 위한 방향 정보를 이용한 최근접 질의 처리 방법을 제안한다. 제안된 질의 처리 방법은 기존의 공간 데이터베이스에서의 유클리디안 거리(Euclidean Distance)를 이용한 최근접 질의 처리 방법에 기반하여 질의 요청자의 이동 방향 정보가 질의 처리 과정에 반영될 수 있도록 확장한 방법이다. 제안된 방법은 질의 처리 과정에서 이동 객체의 방향 정보가 반영될 수 있도록 질의 대상의 방향 정보를 가중치로 환산하여 질의 처리 과정에 적용하였다. 실세계에서는 방향 정보를 단순히 가중치로 환산하였을 때, 차량의 차선 변경처럼 차량의 진행 방향은 동일하지만 일시적인 방향 차이로 인하여 전체 질의 처리 결과에 영향을 주는 상황이 발생한다. 따라서 이러한 상황을 효과적으로 처리하기 위하여 이 논문에서는 방향 정보를 양자화 하여 일시적인 방향 값의 변경에도 적용될 수 있도록 방향 정보 이용 최근접 질의 처리 방법을 확장한 방법을 제안한다.

이 논문의 구성은 다음과 같다. 먼저 2장에서는 최근접 질의와 관련된 기존의 연구들과 이들의 문제점을

살펴본다. 3장에서는 방향 정보를 고려한 최근접 질의 처리를 위한 실세계 모바일 객체를 모델링하며, 4장에서는 제안된 방향 정보를 이용한 최근접 질의 처리 방법을 소개한다. 5장에서는 제안된 방법과 기존의 최근접 질의 처리 방법을 비교 평가하고, 제안된 질의 처리 기법의 실세계 적용 예를 보이며, 마지막으로 6장에서는 결론 및 향후 연구를 제시한다.

## 2. 관련 연구

이 장에서는 공간 데이터베이스에서의 최근접 질의와 이동 객체 환경에서의 최근접 질의들을 소개한다. 또한 기존 기법들의 문제점 및 해결 방안을 살펴본다.

### 2.1 공간 데이터베이스의 최근접 질의

공간 데이터베이스에서는 점질의, 영역질의, 조인질의, 최근접 질의 등 매우 다양한 형태의 질의 유형들이 제공된다. 최근접 질의는 주어진 점에서 가장 가까운 객체를 찾는 질의로써 지금까지 이에 대한 많은 연구들이 수행되어 왔다.

Roussopoulos[4]는 가까운 객체를 빠르게 검색하기 위하여 R-Tree 색인을 이용한 방법을 제안하였다. 이 방법에서는 R-Tree를 구성하는 노드들의 최소경계 사각형 간의 거리 계산 기준을 정의하고, 이 기준에 의하여 탐색 공간을 가지치기 함으로써 최근접 객체를 검색한다. 거리 계산 기준으로 최소 경계 사각형 M의 모서리와 질의 q사이의 최소거리를 의미하는 MINDIST(q, M), M내에 위치하는 임의의 점이 최근접 객체임이 보장될 때, q와 M 사이의 최소 거리를 나타내는 MINMAXDIST(q, M), M의 모서리와 q사이의 최대 거리를 나타내는 MAXDIST(q, M) 등의 세 가지 기준을 제시하였다.

이외의 최근접 질의 처리 방법으로 두 데이터 집합이 주어졌을 때, 하나의 데이터 집합을 질의 데이터로 삼고 다른 데이터 집합에서 최근접 탐색을 행하는 Closet-Pair 집합의 처리 방법이 제안되었다[5][6][7]. 또한 효율적인 k-최근접 질의를 위하여 그룹화 기법을 적용한 방법[8]이 제안되었으며, 최근에는 제약 조건을 통하여 빠르게 최근접 객체를 탐색하는 방법[9]이나, 실세계 도로를 기반으로 한 spatial network database 상에서의 최근접 질의 처리 기법[10]도 제안되고 있다.

2.2 이동객체 환경에서의 최근접 질의

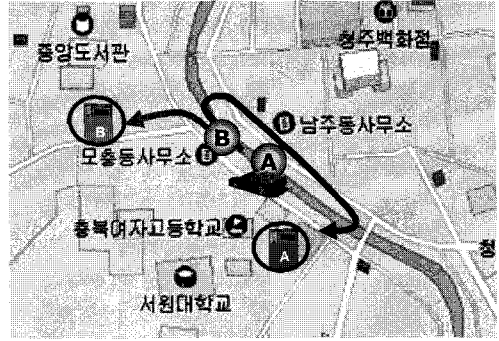
모바일 환경에서는 질의 객체가 시간의 흐름에 따라 연속적으로 이동하기 때문에, 기존의 공간 데이터베이스에서의 최근접 질의 처리 방법을 그대로 적용하였을 경우 많은 비용이 요구된다. 최근 Kollios[11], Benetis[12], Tao[13], 최보윤[14]등은 모바일 환경에서 이동 객체의 이동 특성을 고려한 새로운 형태의 질의 처리 기법들을 제안하였다.

Kollios[11]는 쌍대성 변환(duality transform) 기법을 사용하여 (x, y) 평면상의 이동객체 궤적 선분들을 (속도, y절편) 평면상의 점으로 변환하고, 두 점 사이의 유클리디안 거리를 계산하여 최근접 객체를 탐색하는 기법을 제안하였다. 하지만 이 방법은 모든 객체 타입에 적용 가능하지만, 연속적인 최근접 탐색이 불가능하며 3차원 (x, y, t) 공간상의 선분으로 표현되는 이동 객체 궤적에 대해서는 비효율적이라는 단점이 있다.

Benetis[12]는 객체가 모두 이동하는 경우에 적용 가능한 최근접 탐색 기법과 역(reverse) 최근접 탐색 기법을 제안하였다. 역 최근접 탐색은 자신을 최근접 객체로 가지는 다른 객체를 결과로 반환하며, 이 두 기법의 조합은 쌍방향 최근접 질의 서비스를 가능케 한다. 또한 Tao[13]는 객체가 모두 이동하는 경우 적용될 수 있는 k-CNN 기법을 제안하였다. 이 방법에서 질의점과 데이터 객체점 사이의 거리 계산은 [12]와 동일한 미분 함수를 사용하여 처리되며, 근접원과 수직이등분선은 각각 근접구와 수직 이등분평면으로 확장된다. 최보윤[14]은 연속 최근접 질의 개념과 궤적 최근접 질의 개념에 기반을 둔 CTNN(Continuous Trajectory Nearest Neighbor) 기법을 제안하였다. 이 방법은 변위 계산을 통해 질의 궤적과 가장 가까운 위치를 유지하면서 움직이는 객체를 최근접 객체로 선택한다. 또한 최근접 객체가 변경되는 연속적인 시간을 탐색해내기 위하여 객체 세그먼트들의 변곡점과 교차점 위에서 결과 변경 여부를 비교하여 처리하는 기법이다.

지금까지 살펴본 최근접 질의 연구들은 이동객체가 이동하는 궤적에 연속적으로 요청되는 질의를 효과적으로 처리하기 위한 방법에 초점을 맞추어 왔다. 하지만 <그림 1>에서 처럼 모바일 환경에서는 질의 객체가 시간의 흐름에 따라 지속적으로 이동하는 경우, 질의 요청 시점 A에서의 최근접 객체와 질의 결과 획득 시점 B에서의 최근접 객체의 상태가 달라질 수 있다. 또한 검색된 최근접 대상 객체가 질의 객체 이동 방향

의 반대 방향에 위치한 경우 해당 객체에 접근하기 위한 비용이 오히려 증가되는 문제가 발생한다.



<그림 1> 모바일 환경의 최근접 객체

이 논문은 이동 방향 정보를 갖는 질의 객체의 연속적인 이동 특성을 고려한 DNN (Direction-based Nearest Neighbor) 질의 처리 방법과 방향 정보를 양자화하여 현실 세계에서 발생하는 미세한 방향 오차를 보장하도록 확장한 QDNN (Quantized Direction-based Nearest Neighbor) 질의 처리 방법을 제안한다. DNN 질의 처리 방법은 이동 객체의 방향 정보가 거리 비교 과정에서 반영 될 수 있도록 질의 객체의 방향 정보를 가중치로 환산하여 질의 처리 과정에 반영하는 기법이다. 실세계에서 이동 객체는 전체 경로 상으로는 같은 방향으로 진행하고 있음에도 불구하고, 차로 변경 등과 같이 일시적으로 방향 성분 속성이 달라지는 경우가 발생한다. 이러한 현실 세계의 이동 객체 특성을 보다 적절히 모델링 하기 위하여 이동 방향을 지정된 범위로 양자화 하여 처리하는 QDNN 질의 처리 방법을 제안한다.

3. 최근접 질의 처리를 위한 모바일 객체 모델링

제안된 질의 처리 방법을 설명하기 위하여 모바일 객체를 정의하고, 모바일 객체의 특성을 살펴본다. 또한 이 논문에서 가정하는 몇 가지 제약 조건들을 제시하고, 모바일 추천 시스템에서의 최근접 질의 요청 예를 살펴본다.

3.1 모바일 객체의 특성

모바일 객체는 모바일 환경에 존재하는 임의의 시공간 객체로서 시간이 흐름에 따라 스스로 그 위치를 변화하면서 이동하는 객체를 말한다[15]. 모바일 객체

는 GPS 등의 위치 측위 장비를 장착한 휴대폰 사용자, PDA(Personal Digital Assistant)와 같이 시간의 변화에 따라 객체의 위치 값만이 변경되는 점(point) 객체로 정의된다.

모바일 객체의 이동 형태[16]는 자유 궤적(free trajectory)과 제약 궤적(constrained trajectory)으로 분류할 수 있다. 자유 궤적은  $n$ -차원 공간에서 모바일 객체의 이동에 대한 제약이 없는 경우를 말한다. 예를 들면 “바다에서 이동하는 배”를 들 수 있다. 제약 궤적은 객체의 이동을 강하게 제약하는 경우를 말한다. 예를 들면 “기차의 궤적”과 같은 경우가 있다. 모바일 객체는 일반적으로 자유 궤적을 가지며 이동한다고 정의할 수 있다. 제약 궤적은 자유 궤적의 특별한 경우로 간주될 수 있기 때문이다. 이와 같이 자유롭게 이동하는 모바일 객체의 위치 정보를 획득하는 방법에 따라, 사건 지향(event driven) 시스템과 관측 지향(observation driven) 시스템의 두 가지 형태로 분류할 수 있다. 사건 지향 시스템은 모바일 객체의 속도나 방향의 변화를 자동으로 검출하는 것이 가능한 경우의 시스템을 말하며, 추측 항법 시스템이 이에 속한다. 관측 지향 시스템은 GPS와 같은 센서 시스템을 이용하여 정규적인 시간의 간격에서 정렬된 순서대로 객체의 위치를 획득하는 시스템을 말한다.

이 논문에서는 다음과 같은 가정을 두고 모바일 객체의 위치 정보를 모델링 한다.

- 첫째, 모바일 객체는 2차원 공간에서 이동하는 점 객체를 대상으로 한다.
- 둘째, 질의 객체는 현재 계속 이동하고 있는 상태이며, 모바일 객체의 위치변화 과정을 자유 궤적으로 간주한다.
- 셋째, 모바일 객체의 위치 정보 관리 방법은 GPS, Beacon 등을 이용하는 관측 지향 시스템을 기반으로 한다.
- 넷째, 모바일 환경의 특정 시점에서 이동 중인 객체로부터 고정된 대상 객체를 검색하는 질의 처리 과정을 수행한다.

### 3.2 데이터 모델

제한된 기법은 시간의 흐름에 따라 자신의 위치를 변경하며 연속적으로 이동하는 점 객체(예를 들면, 자동차, 사람, 선박 등)와 특정 위치에 있는 고정 객체(예를 들면, 주유소, 병원, 은행 등)를 대상으로 한다. 객체의 위치 정보는 GPS와 같은 센서 시스템 등의 장치를 통하여 주기적으로 추출되며, 이동 객체 정보

는 주기적으로 샘플링되어 각 시간 time에서의  $\langle oid, time, position(x, y), orient, speed \rangle$  형태로 저장된다. 이 때, oid는 객체의 식별자이고, time은 유효 시간을 나타내는 타임스탬프, position(x, y)는 객체의 위치에 해당되는 좌표를 나타내며, orient와 speed는 방향과 속도 정보를 나타낸다.

모바일 객체의 위치 정보를 모델링 하는 방법에 따라 데이터베이스에 저장되는 데이터 구조 및 연산 처리 방법이 바뀌게 된다. 따라서 이 논문에서는 다음과 같이 모바일 객체를 정의한다.

**【정의 1】** 시간에 따라 위치가 연속적으로 변화하는 점 객체를 모바일 객체라 하고, 모바일 객체는  $MV = \langle T_A, S_A, G_A \rangle$ 로 정의한다.  $T_A, S_A, G_A$ 는 모바일 객체의 시간속성, 공간속성, 일반속성을 나타내며  $T_A, S_A, G_A$ 는 [정의2], [정의3], [정의4]와 같이 정의한다.

**【정의 2】** 모바일 객체의 시간 속성은 유효 시간 간격으로 구성되며  $T_A = \langle vt_s, vt_e \rangle$ 로 정의한다.

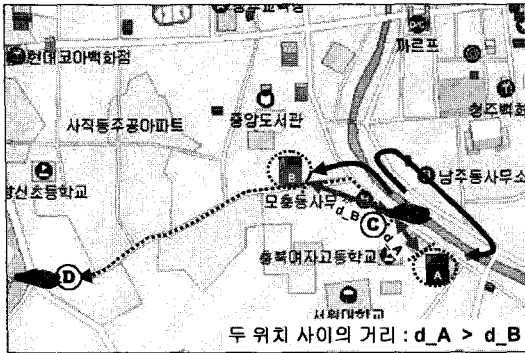
정의 2의  $vt_s$ 는 유효 시작 시간,  $vt_e$ 는 유효 종료 시간을 나타낸다. 이 때,  $vt_s$ 와  $vt_e$ 는 유효 시간의 집합  $T_v$ 의 원소가 된다. 유효 시간은 실제계에서 발생된 시간을 나타내며,  $T_v = \{t_0, t_1, t_2, \dots, t_k, \dots, t_{now}\}$ 이고, 각 원소들은  $t_0 < t_1 < t_2 < \dots < t_k < \dots < t_{now}$ 의 순서를 갖는다.  $t_k = t_{k-1} + 1$ ,  $t_k = t_0 + k$ ,  $k \geq 0$ 인 정수로 정의된다.  $t_{now}$ 는 현재 시간을 의미하는 시간 상수이다. 유효 시간의 도메인은 선형 시간(linear time), 이산 시간(discrete time), 절대 시간(absolute time)이며, 하나의 데이터베이스는 동일한 유효 시간의 주기를 갖는다.

**【정의 3】** 공간 속성은 이동 차량의 특정 시점에서의 위치 좌표 값을 나타내며  $S_A = \langle x, y \rangle$ 로 정의한다.

**【정의 4】** 일반 속성  $G_A$ 는 사용자가 정의하는 속성으로 정의한다.

지금까지 정의한 모바일 데이터를 이용하여 모바일 환경의 추천 시스템을 위한 최근접 질의 연산자를 설명한다. 예를 들어 먼저 임의의 모바일 객체가 C 위

차에서 D 쪽으로 이동하고 있다고 가정하자.



〈그림 2〉 모바일 객체의 이동 과정 예

〈그림 2〉에서 모바일 객체의 위치 정보는 5분 간격으로 데이터 수신기를 통해 입력된다. 이때 “2004년 5월 1일 09시 10분 시점에 차량에서 가장 가까운 곳에 위치한 주유소는 어디인가?”라는 질의가 요청되었을 때, 이 질의는 다음과 같은 구문을 통해 표현할 수 있다.

```
Select oil_id, oil_x, oil_y
From oil_table, car_table
Where NAME = 'CBK2345' and Vs =
      '2004/5/1/09/10' and
      Nearest(car_x, car_y, oil_x, oil_y)
      and SameDirection(car-Direction);
```

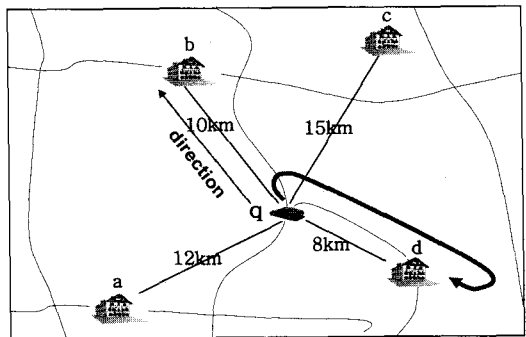
즉, 질의 시점(Vs)과 차량 번호(car\_id)를 입력 값으로 받은 후, 주유소 위치를 검색한다. 차량(car\_x, car\_y)과 주유소 위치(oil\_x, oil\_y)를 가지고 거리를 계산한다. 또한, 차량에 대한 방향 정보와 주유소에 대한 방향 정보를 계산한 후, 차량 방향 정보에 대해 가중치 함수를 적용하여 결과를 도출할 수 있다.

#### 4. 방향 정보를 고려한 최근접 질의 처리

##### 4.1 기존 최근접 질의 처리

일반적인 공간 데이터베이스에서의 최근접 질의 처리는 특정 공간의 주어진 점에서 가장 가까이 위치한 객체들을 유클리디안 거리를 기준으로 정렬하여 k개의 객체들을 찾는다. 예를 들어 사용자가 모니터 위의 어떤 객체나 위치를 선택하면 시스템은 데이터베이스 내

에 저장된 데이터에서 가장 가까운 k 개의 객체를 검색한다. 〈그림 3〉에서 사용자의 위치 q에서 “15km 범위 안에 있는 호텔들을 찾아라”와 같은 질의를 요청한다면, 시스템은 질의 점을 기준으로 가장 가까운 거리에 있는 객체들 a, b, c, d를 검색한다. 특히 차량이 현재 위치 q에서 b방향으로 이동하고 있는 경우, “현재 위치에서 가장 가까운 곳에 위치한 호텔을 검색하라”와 같은 질의 결과는 호텔 d를 질의 결과로 반환될 것이다.

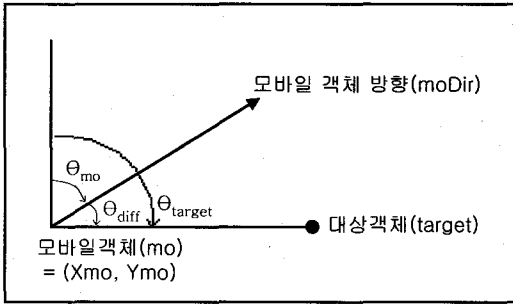


〈그림 3〉 최근접 질의 처리

하지만 모바일 환경에서 질의 요청자는 특정 방향으로 항상 이동을 하기 때문에 차량이 질의를 요청한 시점과 질의 요청결과를 획득하는 시점에서의 결과는 달라질 수 있다. 위의 예에서 차량의 위치에 대한 샘플링 간격이 3분이고, 차량이 질의 위치 q에서 b 방향으로 시속 80km로 이동하는 경우, 질의 획득 시점에서의 질의 결과는 b가 선택되어야 한다. 또한 선택된 대상 객체 b를 접근하기 위해서는 현재 진행 방향과 반대 방향으로 이동하기 위하여 추가적인 시간/거리 비용이 발생한다. 따라서 이러한 문제를 해결하기 위하여 4.2절에서는 질의 처리 시 방향 정보를 반영한 DNN 질의 처리 방법을 제안한다.

##### 4.2 방향 정보를 고려한 최근접 질의 처리

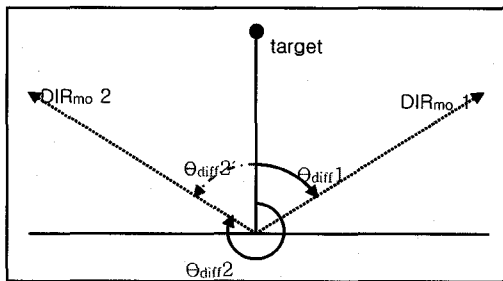
DNN 질의는 모바일 환경에서 연속적으로 이동하는 객체의 방향 성분을 고려하여 최근접 질의 처리를 수행하는 방법이다. 모바일 객체의 방향 성분을 반영하기 위하여 제안된 방법에서는 모바일 객체의 방향 성분을 가중치(weight)로 환산하고, 구해진 가중치를 최근접 질의 처리 과정에서 유클리디안 거리와 곱하여 줌으로써 최종적인 대상 객체의 우선순위를 결정한다.



〈그림 4〉 모바일 객체의 방향 성분

모바일 객체의 방향 성분을 가중치로 변환하기 위하여 먼저 모바일 객체의 이동 방향과 검색 대상이 되는 객체 간의 각도 차이를 구한다. 〈그림 4〉에서와 같이 모바일 객체의 이동 방향을 moDir이라 하고, 검색 대상이 되는 대상 객체를 target이라 하자. 또한 기준 축으로부터 모바일 객체의 이동 방향 성분은  $\theta_{mo}$ , 기준 축으로부터 검색 대상 객체와의 각도는  $\theta_{target}$ 이라 할 때, 모바일 객체의 이동 방향과 검색 대상 객체간의 방향 차는  $\theta_{diff}$ 라고 정의 된다. 따라서  $\theta_{diff}$  값을 이용하여 최근접 질의 처리를 위한 가중치를 계산할 수 있다.

방향 정보를 가중치로 환산하기 위해서는  $\theta_{mo}$ 나  $\theta_{target}$ 와 같이  $0^\circ \sim 360^\circ$ 사이의 정확한 각도 보다는 두 객체 각도 간의 차이인  $\theta_{diff}$  값을 이용한다. 〈그림 5〉와 같이 모바일 객체와 검색 대상 객체 간의 각도 차이는  $0^\circ \sim 360^\circ$ 사이의 값으로 산출될 수 있다. 하지만  $0^\circ \sim 180^\circ$ 사이의 값과  $180^\circ \sim 360^\circ$ 사이의 값은 대칭관계가 있기 때문에 가중치 계산의 편의성을 위하여  $\theta_{diff1}$ 와 같이 두 객체 사이의 각도 차이가  $180^\circ$ 이상일 경우는  $\theta_{diff2}$ 와 같이 각도를 변경 시켜준다. 따라서  $\theta_{diff}$ 는  $0 \leq \theta_{diff} \leq 180^\circ$ 의 범위 내에 존재하며, 단순히 방향 각도의 차를 이용하기 때문에 +/- 부호는 무시된다.



〈그림 5〉 모바일 객체와 대상 객체의 방향

위와 같은 방법으로 모바일 객체의 이동 방향과 검색 대상 객체의 각도 차이  $\theta_{diff}$ 는 다음 (식 1)에 의해 계산되며,  $(\theta_{diff} > 180^\circ)$ 인 경우는 (식 2)에 의해  $\theta_{diff}$  값이  $0 \leq \theta_{diff} \leq 180^\circ$ 의 값을 갖도록 조정된다.

$$\theta_{diff} = |\theta_{target} - \theta_{mo}| \quad (식 1)$$

$$\theta_{diff} = 360^\circ - \theta_{diff} \quad (식 2)$$

여기서  $\theta_{diff}$ 값이 0에 가까울수록 모바일 객체의 이동 방향과 대상 객체는 같은 방향이다. 또한 가중치 값의 범위는  $W_{min} \sim W_{max}$  사이의 값을 가지며,  $\theta_{diff}$ 가 0에 가까울수록 가중치는  $W_{min}$ 에 가까운 값을 갖는다. 즉,  $\theta_{diff}$ 와  $W_{max}$ 는 비례 관계에 있으며,  $\theta_{diff}$ 가 작을수록 더 작은 가중치를 곱하기 때문에 최근접 거리를 계산하는데 사용되는 유클리디안 거리는 가까워지도록 조정된다. 지금까지의 가중치 산출을 위해 정의된 기호들을 정리하면 다음 〈표 1〉과 같다.

〈표 1〉 기호 정의

기호	설 명
$\theta_{mo}$	모바일 객체의 이동 방향에 대한 각도 값
$\theta_{target}$	모바일 객체를 중심으로 검색 대상 객체의 각도 값
$\theta_{diff}$	모바일 객체와 대상 객체 간의 각도 차
$W_{min}$	가중치가 갖는 최소 값
$W_{max}$	가중치가 갖는 최대 값
$W_{diff}$	최대 가중치( $W_{max}$ )와 최소 가중치( $W_{min}$ )의 차

방향 정보에 대한 가중치 Weight는  $\theta_{diff}$ 에 의해 산출되며,  $0^\circ \sim 180^\circ$  사이의 값을 갖는  $\theta_{diff}$ 는 다음과 같은 (수식 3)에 의해 가중치 값으로 환산된다.

$$Weight = 1 - \frac{(180 - \theta_{diff})}{180} \quad (식 3)$$

여기서 구해진 Weight 값의 범위는  $0 \leq Weight \leq 1$ 이며, Weight가 임의의 값의 범위인  $W_{min}$ 과  $W_{max}$  사이의 값을 갖도록 (수식 3)을 일반화하면 다음의 (식 4)가 유도된다.

$$Weight = W_{max} - \frac{(180 - \theta_{diff})}{180} \times W_{diff} \quad (식 4)$$

따라서 방향 정보를 고려한 최근접 질의는 위에서 구한 Weight 값을 기존의 최근접 질의 처리 과정의 유클리디안 거리에 곱해 줌으로써 처리된다.

다음의 <알고리즘 1>은 방향 정보를 고려한 최근접 질의 처리 과정을 보여준다. *findNearest* 알고리즘은 모바일 객체 moID와 임의 시점 ts를 입력 값으로 받는다. 입력 값을 사용하여 차량의 속성 정보가 저장된 moHistory 테이블에서 입력된 moID를 갖는 차량을 검색한다. 또한 주유소의 속성 정보가 저장된

targetHistory 테이블에 입력된 주유소 정보를 검색한다. 차량과 주유소 사이의 거리를 계산하고 방향 정보를 계산하고, 방향 정보에 대한 가중치를 계산한다. 최종적으로 차량의 진행 방향과 유사한 가장 근접한 위치의 주유소를 검색한다.

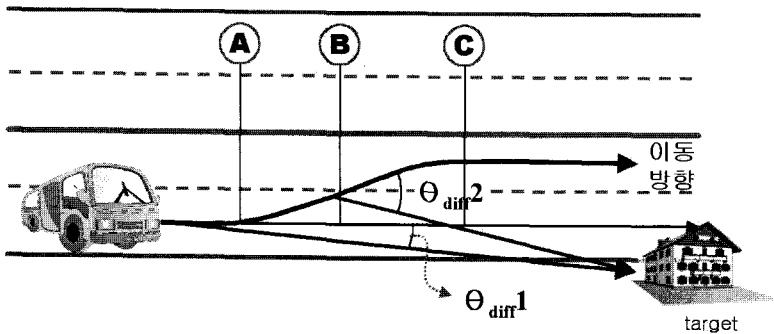
<알고리즘 1> DNN 질의 처리 알고리즘

```

Algorithm findNearest(moID, ts)
input moID      : 임의의 모바일 객체, ts : 질의 시점
output List NL  : 최근접 질의 리스트
Begin
  Step 1. 질의 시점 ts에서의 moID 위치, target 위치 검색
    1.1 moID 위치 : (moID, ts, moX, moY, moDirection, moSpeed)
    1.2 target 위치 : (targetID, targetX, targetY)
  Step 2. moID 위치와 target 위치 간의 거리 Distance 계산
  Step 3. moID 와 target 위치의 방향 정보 검색
    3.1 moID 위치에 대한 이동 방향 각도 검색
    3.2 삼각 함수를 사용하여 target 위치에 대한 각도 계산
  Step 4. 단계 2의 방향 정보에 따라 가중치 계산
    4.1 if ( $\theta_{mo} > \theta_{target}$ ) then
      // 모바일 객체 진행 방향 각도가 대상 객체 각도보다 크다면
       $\theta_{diff} = \theta_{mo} - \theta_{target}$ 
    else then // target이 모바일 객체의 이동 방향 각도보다 크다면
       $\theta_{diff} = \theta_{target} - \theta_{mo}$ 
      if ( $\theta_{diff} > 180$ ) then // 각도가 180°보다 크다면
         $\theta_{diff} = 360 - \theta_{diff}$  // 각도 차 ( $\theta_{diff}$ )
    4.2  $Weight = W_{max} - ((180 - \theta_{diff}) / 180) * W_{diff}$ 
  Step 5. 거리와 가중치를 기반으로 최근접 질의 리스트 작성
    5.1  $Estimation = Distance * Weight$ 
    5.2 Estimation을 NL에 추가
    5.3 NL을 Estimation 값을 기준으로 정렬
  Step 6. Return NL
End
    
```

DNN 질의 처리 알고리즘은 질의 처리 시 실제계의 모바일 객체의 방향 정보를 고려하여 최근접 질의 처리를 수행한다. 하지만 실제계에서는 모바일 객체가

같은 방향으로 진행하고 있음에도 불구하고 일시적으로 약간의 진행 방향 오차가 발생하는 경우가 빈번하게 발생한다. 예를 들어, <그림 6>과 같이 차량이 진



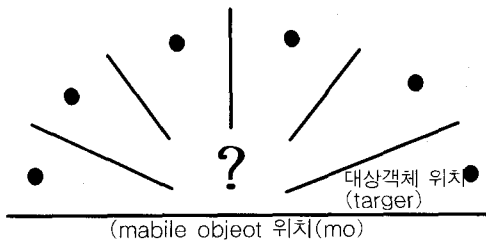
<그림 6> 이동 객체 진행 방향에 따른  $\theta_{diff}$ 의 변화

행 중에 차선을 변경하는 경우, 차량의 전체적인 이동 방향은 동일함에도 불구하고 이동 차량과 검색 대상 객체간의 방향 각도의 차이는 달라진다.

〈그림 6〉에서 차량의 진행 방향은 같음에도 불구하고 A, B, C 시점에서의 제안된 DNN 질의 처리 방법에 의한 검색 결과는 다르게 나타날 수 있다. 따라서 모바일 객체가 이동 중에 발생하는 약간의 방향 정보의 오차를 극복하기 위하여 이 논문에서는 이동 객체와 대상 객체간의 방향 정보를 양자화 하여 처리하도록 DNN 질의 처리 알고리즘을 확장한 QDNN 질의 처리 알고리즘을 제안한다.

**4.3 양자화 된 방향 정보를 고려한 최근접 질의 처리**

QDNN 질의 처리 방법은 모바일 객체의 진행 방향에 따른 오차를 극복하도록 DNN 질의 처리 방법을 확장하였다. DNN 질의 처리 알고리즘에서 설명한 바와 같이 모바일 객체의 진행 방향과 대상 객체 간의 각도 차이  $\theta_{diff}$ 는 0 ~ 180° 사이의 값을 갖는다. 하지만 실제계에서 모바일 객체는 동일한 진행 방향을 가짐에도 불구하고, 일시적으로 방향 성분 값이 달라지는 현상이 발생한다. 이러한 문제를 해결하기 위하여 0 ~ 180° 사이의 방향 값을 〈그림 7〉과 같이 양자화 하여 일정한 간격 내의 방향들은 같은 각도, 즉 같은 가중치를 갖도록 한다.



〈그림 7〉 방향 정보 양자화

예를 들어 30° 간격으로 양자화 시킬 경우, 0~30° 사이의 각도 차는 같은 가중치를 갖도록 한다. 이때 양자화 한 각도를 QD (Quantized Degree)라 하고, 양자화된 각도의 분할된 수를 QN (Quantized Number)이라고 정의한다. 따라서 양자화 각도와 양자화 수는 (식 5)와 (식 6)에 의해 계산된다.

$$QD = \frac{180}{QN} \tag{식 5}$$

$$QN = \left\lfloor \frac{180}{QD} \right\rfloor \tag{식 6}$$

(식 6)을 이용하여 양자화된 각도에 의한 가중치 계산은 다음 식에 의해 계산된다.

$$Weight = 1 - \frac{QN - (\theta_{diff}/QD)}{QN} \tag{식 7}$$

여기서 구해진 Weight 값의 범위는  $0 \leq Weight \leq 1$  이며, Weight가  $W_{min}$ 과  $W_{max}$  사이의 값을 갖도록 (식 7)을 일반화하면 다음 (식 8)을 유도할 수 있다.

$$Weight = W_{max} - \frac{QN - |\theta_{diff}/QD|}{QN} \times W_{diff} \tag{식 8}$$

(식 8)에서 구한 Weight 값을 사용하여 최근접 질의 처리를 수행하도록 수정한 알고리즘이 〈알고리즘 2〉에 제시되었다. 〈알고리즘 2〉의 findQNearest 알고리즘은 이동 객체 moID와 임의의 시점 ts를 입력 값으로 받는다. 입력 값을 사용하여 차량의 속성 정보가 저장된 moHistory 릴레이션에서 입력된 moID를 갖는 차량을 검색한다. 그리고 주유소의 속성 정보가 저장된 targetHistory 테이블에 입력된 주유소 정보를 검색한다. 차량과 주유소 사이의 거리를 계산하고 방향 정보를 검색한다. 또한 각도 차이의 오차를 줄이기 위해 방향 정보의 양자화를 통해 가중치를 계산한다. 그러므로 차량과 주유소 사이에 가까운 거리에 위치하고 차량이 이동하는 방향에 위치한 주유소 위치를 찾아 준다.

**5. 실험 및 성능 분석**

이 장에서는 특정 예를 사용한 결과 화면을 통하여 제안한 시스템의 구현 결과를 보여준다. 예제로 사용된 데이터는 2003년 2월 7일에 강동구를 이동한 차량의 위치 정보이다. 이러한 데이터들은 5.2에서 제시하는 모바일 객체 데이터베이스 구조 형태로 저장된다. 또한 실험 및 평가에서는 기존 최근접 질의와 DNN 질의와 QDNN 질의를 비교, 평가 한다.



〈알고리즘 2〉 QDNN 질의 처리 알고리즘

---

```

Algorithm findQNearest(moID, ts)
input  moID   : 입의의 모바일 객체
       ts     : 질의 시점
output List NL : 최근접 질의 리스트

Begin
Step 1. 질의 시점 ts에서의 moID 위치, target 위치 검색
    1.1 moID 위치 : (moID, ts, moX, moY, moDirection, moSpeed)
    1.2 target 위치 : (targetID, targetX, targetY)
Step 2. moID 위치와 target 위치 간의 거리 Distance 계산
Step 3. moID 와 target 위치의 방향 정보 검색
    3.1 moID 위치에 대한 angle 검색
    3.2 삼각 함수를 사용하여 target 위치에 대한 angle 검색
Step 4. 단계 2의 방향 정보에 따라 가중치 계산
    4.1 if ( $\theta_{mo} > \theta_{target}$ ) then
        // 모바일 객체 각도가 대상 객체 각도보다 크다면
         $\theta_{diff} = \theta_{mo} - \theta_{target}$ 
      else then // target이 mobile object보다 크다면
         $\theta_{diff} = \theta_{target} - \theta_{mo}$ 
        if ( $\theta_{diff} > 180$ ) then // 각도가 180°보다 크다면
           $\theta_{diff} = 360 - \theta_{diff}$ 
    4.2  $Weight = W_{max} - ((QN - (\theta_{diff} / QD)) / QN) * W_{diff}$ 
Step 5. 거리와 가중치를 기반으로 최근접 질의 리스트 작성
    5.1  $Estimation = Distance * Weight$ 
    5.2 Estimation을 NL에 추가
    5.3 NL을 Estimation 값을 기준으로 정렬
Step 6. Return NL
End
    
```

---

5.1 구현 환경

모바일 추천 시스템을 위한 최근접 질의 처리는 Java JDK 1.3을 사용하여 구현하였으며, 차량의 일반 속성 데이터 관리를 위한 DBMS로는 Microsoft SQL Sever 2000을 사용하였다.

5.2 모바일 데이터베이스

실험에서 사용된 모바일 객체 데이터베이스의 테이블 구조는 다음과 같다. 특정 시간 간격으로 샘플링되는 모바일 객체 정보와 대상 객체 정보는 다음과 같이 2개의 테이블로 저장 관리한다.

moHistory(moID, ts, moX, moY, moDirection, moSpeed)  
 targetHistory(targetID, targetX, targetY)  
 moHistory 테이블은 모바일 객체의 시공간 속성 정보를 관리하기 위한 것이며 targetHistory 테이블은 대상 객체의 시공간 속성 정보를 관리한다. 이 논문에서는 실세계에서 빈번하게 발생하는 최근접 질의를 처리하기 위하여 모든 모바일 객체들의 이력 정보를 moHistory 테이블에 저장, 관리한다. 또한 모바일 객체의 이동 정보 갱신을 위한 추가적인 갱신 전략은 이 논문에서는 고려하지 않았다. 각각의 테이블의 상세한 속성 정보는 다음과 같다. 모바일 객체 정보 테이블 moHistory는 〈표 2〉와 같은 속성 값들을 갖는다.

〈표 2〉 모바일 객체 테이블

속 성	moID	ts	moX	moY	moDirection	moSpeed
자료형	int	string	double	double	int	int
설 명	키	질의유효시간	x 좌표	y 좌표	이동 방향 (°)	이동 속도 (km/h)

각각의 모바일 객체들에 대한 이동 위치에 관련된 질의를 수행하기 위해 이동 방향과 이동 속도가 저장되며, 이 정보가 어느 시점부터 적용되는 지를 나타내는 시간 정보와 위치 정보를 포함한다.

targetHistory는 대상 객체를 표현하기 위한 속성 값들을 가진다. targetHistory 테이블은 현재 시공간 객체들의 위치 정보를 관리한다.

〈표 3〉 대상 객체 테이블

속 성	targetID	targetX	targetY
자료형	int	double	double
설 명	키	x 좌표	y 좌표

〈표 3〉은 대상 객체들에 대한 이동 위치와 관련된 질의를 수행하기 위해 유지된다. 대상 객체에 대한 방향 각도는 대상 객체의 위치 정보에 대한 삼각함수를 이용하여 각도 정보를 계산할 수 있다. 이 각도 정보는 모바일 객체와 대상 객체간의 방향이 같은 방향인지 다른 방향인지를 알 수 있는 중요한 정보이다.

5.3 질의 수행 결과

이 절에서는 제안된 질의 기법이 적용된 모바일 객체 관리 시스템 응용에서 질의를 수행함으로써 기존 질의, DNN 질의, QDNN 질의에 대해 비교해 본다.

실험에 사용된 차량과 주유소 정보는 〈표 4〉, 〈표 5〉와 같다.

〈표 4〉 차량 정보

차량 이름	시간 정보	x 좌표	y 좌표	속도	방향
SL81BA3578	2003-02-07-16-32	211328.4900	448304.2900	34	111

〈표 5〉 주유소 정보

주유소 이름	X 좌표	Y좌표
강서 주유소	210872.8125	447541.0938
성내 주유소	211213.1563	448133.7500
강남 주유소	211846.6094	448167.7188
한국 주유소	212015.0313	448291.6250
개신 주유소	212561.6875	448233.3438
명일 주유소	212811.1719	449728.4063
강동 주유소	211818.7031	449928.1875
천호 주유소	211207.0156	448713.7500
현대 주유소	210872.8125	448504.8750
서울 주유소	210807.1875	448272.1250

이때 “2003년 2월 7일 16시 32분에 강동구 지역을 이동하는 차량 SL81BA3578의 위치에서 가장 가까운 주유소는 어느 것인가?”라는 질의가 요청되었을 때 유클리디안 거리와 DNN 질의 처리, QDNN 질의 처리 결과는 〈표 6〉, 〈표 7〉과 같다. 실험에서 사용된 가중치 값의 범위는  $0.5 \leq Weight \leq 1$  이며, QDNN에서의 양자화 각도  $QD = 15$ 이다.

〈표 6〉 DNN 질의 처리 결과

주유소 이름	실제 거리	$\theta_{diff}$	$W_{DNN}$	$DNN_{distance}$
명일 주유소	178.8309	178	1.00	178.8309
개신 주유소	221.2326	30	0.58	128.3149
한국 주유소	260.6541	7	0.52	135.5401
강동 주유소	316.2747	108	0.80	253.0197
강남 주유소	317.8126	12	0.53	168.4406
천호 주유소	425.1001	48	0.63	267.8132
성내 주유소	476.8802	26	0.57	271.8217
현대 주유소	515.8754	90	0.75	386.9065
서울 주유소	546.2978	38	0.61	333.2416
강서 주유소	632.8836	21	0.56	354.4148

〈표 6〉에서와 같이 실제 거리상으로는 명일 주유소가 가장 가까운 위치에 있지만 DNN 질의 결과로는 개신주유소가 반환되는 것을 볼 수 있다. 또한 〈표 7〉에서 QDNN 질의 처리 방법에 의해 계산할 경우, 천호주유소와 성내주유소의 순위가 바뀐 것을 확인할 수 있다.

〈표 7〉 QDNN 질의 처리 결과

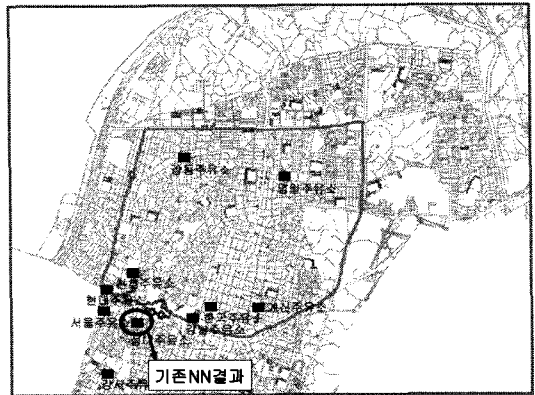
주유소 이름	실제 거리	$\theta_{diff}$	$W_{QDNN}$	$QDNN_{distance}$
명일 주유소	178.8309	178	0.96	171.6776
개신 주유소	221.2326	30	0.58	128.3149
한국 주유소	260.6541	7	0.50	130.3270
강동 주유소	316.2747	108	0.79	249.8570
강남 주유소	317.8126	12	0.50	158.9063
천호 주유소	425.1001	48	0.63	267.8130
성내 주유소	476.8802	26	0.54	257.5153
현대 주유소	515.8754	90	0.75	386.9065
서울 주유소	546.2978	38	0.58	316.8527
강서 주유소	632.8836	21	0.54	341.7571



〈그림 8〉 DNN 질의 처리 결과 화면

〈그림 8〉은 DNN 질의 처리 방법에 의한 질의 수행 결과를 보여준다.

〈그림 8〉에서와 같이 질의 시점에서 차량으로부터 가장 가까이 위치한 주유소는 "명일주유소"이지만, DNN 질의 처리 과정에서 "개신주유소"가 선택되어지는 것을 볼 수 있다. 위 〈그림 8〉과 같이 모바일 환경에서는 질의 요청자 및 질의 대상 객체가 시간의 흐름에 따라 그 위치가 수시로 변화한다. 따라서 위의 예에서 기존의 최근접 질의 처리 방법에 의한 질의 결과를 이용할 경우 "명일주유소"가 선택되어 지고, 차량이 현재의 위치와 이동 방향에서 명일 주유소로 이동하려 할 경우는 도로의 U-턴 가능한 곳까지 이동을 하여 다시 돌아오거나 "명일주유소"를 가기위한 다른 도로로 우회하여 가야한다. 하지만 제안된 방법에 의해 얻어진 질의 처리 결과인 "개신주유소"의 경우, 질의 요청시의 거리로는 "명일주유소"보다 멀리 떨어져 있지만 모바일 객체의 이동 방향과 같은 방향에 위치하기 때문에 추가적인 시간/거리의 비용을 소모하지 않고 접근할 수 있다. 따라서 이러한 모바일 객체의 이동 특정 즉, 객체의 현재 위치뿐만 아니라 이동 방향을 고려할 경우 보다 실세계의 사용자의 요구에 적합한 질의 처리 결과를 얻을 수 있다.



〈그림 9〉 다른 형태의 모바일 객체의 이동

〈그림 9〉는 모바일 객체의 다른 형태의 이동 경로를 보여 준다. 〈그림 9〉의 경우, 제안된 질의 처리 방법에 의한 결과는 〈표 8〉과 같다.

〈표 8〉에서와 같이 기존의 최근접 질의 결과에 의한 5개의 최근접 리스트는 [성내, 강남, 천호, 현대, 서울]이지만, DNN/QDNN 질의 처리 방법에 의해서는 [성내, 천호, 현대, 서울, 강남] 순으로 질의 결과 리스트가 변화함을 알 수 있다.

〈표 8〉 질의 처리 결과

주유소 이름	실제 거리	$\theta_{diff}$	WDNN	DNN <sub>distance</sub>	WQDNN	QDNN <sub>distance</sub>
성내 주유소	76.8981	53	0.65	49.9837	0.63	48.4458
강남 주유소	112.2309	177	0.99	111.1086	0.96	107.7416
천호 주유소	121.2312	30	0.59	71.5264	0.59	71.5264
현대 주유소	156.7305	2	0.51	79.9325	0.50	78.3652
서울 주유소	157.3261	23	0.57	89.6758	0.55	86.5293
한국 주유소	160.2743	160	0.95	152.0000	0.92	147.4523
강서 주유소	235.8452	73	0.71	167.4501	0.67	158.0163
개신 주유소	286.7212	161	0.95	272.3851	0.92	263.7835
강동 주유소	440.5295	83	0.73	321.5865	0.71	312.7759
명일 주유소	525.2683	117	0.83	435.9727	0.80	420.2146

6. 결론

참고문헌

모바일 환경에서 자주 사용되는 질의 처리 기법 중 하나인 최근접 질의는 질의 요청자의 위치를 기준으로 가장 가까운 곳에 위치한 객체를 검색하는 질의이다. 모바일 환경에서는 질의 객체가 연속적으로 이동한다는 특성 때문에 질의 요청자의 위치에 따라 연속적인 질의 요청이 발생하기도 하며, 질의 객체의 이동 방향 속성에 따라 제공되어야 할 질의 결과가 달라질 수 있다.

이 논문에서는 모바일 환경에서 질의 요청자의 이동 방향을 고려한 최근접 질의 처리 방법을 제안하였다. 제안된 방법은 질의 처리 과정에서 모바일 객체의 방향 정보가 반영 될 수 있도록 질의 요청자의 방향 정보를 가중치로 환산하여 질의 처리 과정에 적용하는 기법을 제안하였다. 또한 실세계의 모바일 객체의 이동 특성 상 같은 방향으로 진행하는 과정에서 발생할 수 있는 약간의 방향 오차를 보정하기 위하여 가중치를 산출하는 과정에서 반영되는 방향 정보를 양자화하여 최근접 질의를 처리하는 방법을 제시하였다.

제안된 최근접 질의 기법을 교통정보 시스템, 관광 정보 시스템, 물류관리 시스템, 소방안전 시스템과 같은 응용 시스템에 적용가능하다. 향후에는 모바일 객체의 방향 속성뿐만 아니라 객체의 이동 속도, 객체가 이동하는 도로 정보 등을 이용하여 실세계 환경을 보다 효과적으로 처리하기 위한 연구를 진행할 것이다.

- [1] 김종혁, "첨단 교통관리 시스템", 정보과학회지, 제16권 6호, 1998, pp. 5~13.
- [2] 안승범, "안전도 향상을 위한 첨단 화물운송 시스템(CVO)의 서비스와 기술", 정보과학회지, 제16권 6호, 1998, pp.30~35.
- [3] 이승룡, 홍영래, 김형일, 배수강, 최대순, "첨단 대중교통 시스템", 정보과학회지, 제16권 6호, 1998, pp.23~29.
- [4] N. Roussopoulos, S. Kelley, F. Vincent, "Nearest Neighbor Queries", SIGMOD Conference 1995, pp.71~77.
- [5] A. Corral, Y. Manolopoulos, Y. Theodoridis, M. Vassilakopoulos, "Closest Pair Queries in Spatial Database", SIGMOD Conference 2000, pp.189~220.
- [6] C. A. Lang, A. K. Singh, "Accelerating High-Dimensional Nearest Neighbor Queries", SSDBM 2002, pp.109~129.
- [7] J. Zhang, N. Mamoulis, D. Papadias, Y. Tao, "All-Nearest Neighbor Queries in Spatial Database", Technical Report, February 2002.
- [8] C. Yu, B. C. Ooi, K. L. Tan, H. V. Jagadish, "Indexing the Distance : An Efficient Method to KNN Processing", VLDB 2001, pp.421~430.

- [9] H. Ferhatosmanoglu, I. Stanoi, D. Agrawal, A. E. Abbadi, "Constrained Nearest Neighbor Queries", SSTD 2001, pp.257~278.
- [10] D. Papadias, J. Zhang, N. Mamoulis, Y. Tao, "Query Processing in Spatial Network Databases", VLDB 2003, pp.9~12.
- [11] G. Kollios, D. Gunopulos, V. J. Tsotras, "Nearest Neighbor Queries in a Mobile Environment", Spatio-Temporal Database Management 1999, pp.119~213.
- [12] R. Benetis, C. S. Jensen, G. Karciuskas, S. Saltenis, "Nearest Neighbor and Reverse Nearest Neighbor Queries for Moving Objects", IDEAS 2002, pp.44~45.
- [13] Y. Tao, D. Papadias, Q. Shen, "Continuous Nearest Neighbor Search", VLDB 2002, pp.287~298.
- [14] 최보윤, 지정희, 김상호, 류근호, "이동객체 쿼리에 대한 최근접 질의", 개방형 지리정보시스템 학회 추계 학술대회, 2003, pp. 169~174.
- [15] O. Wolfson, P. Sistla, B. Xu, J. Zhou, S. Chamberlain, N. Rische, and Y. Yesha, "Tracking Moving Objects Using Database Technology in DOMINO", Proc. of NGITS '99, The 4th Workshop on Next Generation Information Technologies and Systems, Zikhron-Yaakov, Israel, 1999, pp.112~119.
- [16] M. A. Nascimento, J. R. Silva, Y. Theodoridis, "Evaluation of Access Structures for Discretely Moving Points", Proceedings of International Workshop on Spatio-Temporal Database Management, STDBM'99, September 1999, pp. 171~188.



이응재

1994년 충북대학교 컴퓨터과학과 졸업 (학사)  
 1996년 충북대학교 대학원 전자계산학과 졸업 (석사)  
 2001년~현재 충북대학교 대학원 전자계산학과 박사과정

관심분야 : 시공간 데이터베이스, 이동객체 데이터베이스, LBS, 지리정보 시스템

정영진



이동객체 색인,

지리정보 시스템

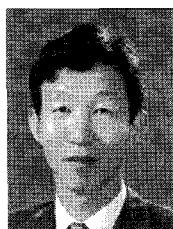
2000년 충북대학교 전자계산학과 졸업 (학사)  
 2002년 충북대학교 대학원 전자계산학과 졸업 (이학석사)  
 2003년~현재 충북대학교 대학원 전자계산학과 박사과정  
 관심분야 : 이동객체 데이터베이스,



최현미

2002년 한밭대학교 컴퓨터공학과 졸업 (학사)  
 2002년~현재 충북대학교 전자계산학과 석사과정

관심분야 : 공간데이터베이스, 이동객체 데이터베이스, LBS



류근호

1976년 숭실대학교 전산학과 (학사)

1980년 연세대학교 대학원 전산 전공 (석사)

1988년 연세대학교 대학원 전산 전공 (박사).

1976년~1986년 육군군수 지원사 전산실(ROTC 장교), 한국전자통신연구원(연구원), 한국방송대 전산학과(조교수) 근무.

1989년~1991년 Univ. of Arizona Research Staff (TempIS 연구원, Temporal DB).

1986년~현재 : 충북대학교 컴퓨터과학과 교수

관심분야 : 시간데이터베이스, 시공간 데이터베이스, Temporal GIS, 지식기반 정보검색시스템, 데이터마이닝, 객체와 지식베이스 시스템, 데이터베이스 보안 및 바이오인포매틱스



이성호

1997년 충북대학교 컴퓨터과학과 졸업 (학사)

2000년 충북대학교 대학원 전자계산학과 졸업 (석사)

2000년~현재 한국전자통신연구원 텔레매틱스연구단 선임연구원

관심분야 : 시공간 데이터베이스, GIS, LBS 등