

이동하는 차량들간 최근접 질의 처리 기법

Dynamic Nearest Neighbor Query Processing for Moving Vehicles

이 명 수*
(Myong-Soo Lee)

심 규 선**
(Kyu-Sun Shim)

이 상 근***
(SangKeun Lee)

요 약

세 대 이상의 빠르게 이동하는 차량들은 때론 서로 모이기 위해 모일 장소를 알아야 될 필요가 있다. 이때 각 차량들은 다른 속도를 가지고 있으며, 여러 대의 차량이 짧은 거리를 이동해 빠르게 모이게 하기 위한 방법이 필요하다. 이러한 방법은 그룹기반의 최근접 질의로서 기존의 연구가 진행되어 왔으나, 기존 연구는 이동하지 않는 객체들을 다루고 있어 움직이는 차량에 적용하기엔 어려운 점이 있다. 본 논문에서는 이동하는 차량들에게 효율적인 차량간 최근접 질의 처리 기법을 제안한다. 본 기법은 각 차량의 움직임은 방향과 속도를 기반으로 모든 차량이 최소 시간에 모일 수 있는 최근접 질의점을 찾을 수 있다. 본 기법은 효율적으로 질의점의 그룹을 표현하는 센트로이드를 통해 그룹기반의 최근접을 계산한다. 실험 결과는 제안하는 기법이 움직이는 차량의 최근접 질의 처리에 효율적임을 보여준다.

Abstract

For three and more rapidly moving vehicles, they want to search the nearest location for meeting. Each vehicle has a different velocity and a efficient method is needed for shifting a short distance. It is observed that the existing group nearest-neighbor query has been investigated for static query points; however these studies do not extend to highly dynamic vehicle environments. In this paper, we propose a novel Dynamic Nearest-Neighbor query processing for Multiple Vehicles (DNN_MV). Our method retrieves the nearest neighbor for a group of moving query points with a given vector and takes the direction of moving query points with a given vector into consideration for DNN_MV. Our method efficiently calculates a group nearest neighbor through a centroid point that represents the group of moving query points. The experimental results show that the proposed method operates efficiently in a dynamic group nearest neighbor search.

Key words: Group nearest neighbor, moving vehicles, dynamic environments, location-based services, centroid

I. 서 론

위치기반 서비스(Location-based Service, LBS)는

GPS와 모바일 단말기 무선 통신의 발달과 함께 많은 발전을 해오고 있다 [1, 2]. 위치기반 서비스는 개인 모바일 사용자나 차량 네비게이션에 탑재되어 많

† 이 연구에 참여한 연구자는 '2단계 BK21사업'의 지원비를 받았다.

* 주저자 : 고려대학교 컴퓨터통신공학부 박사과정

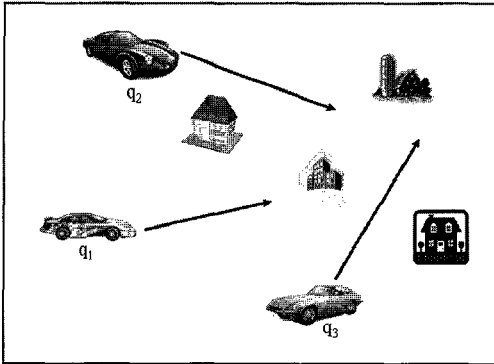
** 공저자 : 고려대학교 컴퓨터통신공학부 석사과정

*** 공저자 및 교신저자 : 고려대학교 컴퓨터통신공학부 부교수

† 논문접수일 : 2009년 12월 2일

† 논문심사일 : 2009년 1월 18일

† 게재확정일 : 2009년 1월 20일



<그림 1> 탐색 예제
<Fig. 1> A search example

은 어플리케이션을 만들어 내고 있으며, 텔레메틱스의 주요 콘텐츠중 하나로서 다양한 연구가 진행되고 있다. 위치 기반서비스 중에서 중요한 질의중 하나로써 최근접 질의는 대부분의 공간 데이터베이스 [3, 4]에서 가장 중요한 질의중 하나이다. 최근접 질의중 하나의 예는 다음과 같다. 차량으로 이동중인 A씨는 주위에 가장 가까운 주유소를 찾기 위하여 주유소의 위치를 모두 가지고 있는 중앙 위치기반 서비스 서버에 자신의 위치를 알려주어, 중앙 서버에서 거리계산을 해 가장 가까운 주유소의 위치를 A씨의 단말기에 전송해준다.

질의점이 여러개인 경우 그룹 기반의 최근접 질의(Group Nearest Neighbor, GNN)가 있다 [5]. 본 질의는 그룹 기반의 최근접은 여러 개의 질의점과의 모든 거리 합이 가장 짧은 객체를 결과값으로 가지는 게 목적이다. 즉, 여러 대의 차량이 모일만한 장소를 섭외할 때 유용하게 사용될 수 있다. 하지만, 그룹 기반의 최근접 질의는 현재 시점에서 가장 가까운 최근접만을 찾는다는 한계가 있다. 이는 역동적으로 이동하는 도로위의 차량에 적용하기에는 한계가 있다. 본 논문에서 해결하려고 하는 어플리케이션의 예제는 그림 1과 아래 예제와 같다.

예제: 세대의 차량이 q_1, q_2, q_3 의 위치에서 각각 주어진 벡터 속도를 가지고 이동중이다. 세대의 차량의 운전자는 세 명이 모일만한 레스토랑을 찾아보려 하고 있다. 세 명의 운전자는 향후 자신의 스케줄에 맞추어 미팅장소를 정하기 위해 지금 상황에서

가장 가까운 미팅 장소가 아닌 지금부터 향후 1시간 내에서 시시각각으로 가장 가까운 레스토랑의 리스트를 받아 원하는 시간대에 원하는 장소를 섭외하길 원한다.

본 논문에서는 위 예제에 맞는 이동하는 차량들간 최근접 질의 처리 방법론(Dynamic Nearest Neighbor for Multiple Vehicles, DNN_MV)을 제안하고자 한다. 기존의 그룹 기반의 최근접 질의로서는 현재 상황에서 가장 가까운 최근접만 찾을 수 있으므로, 새로운 기법을 적용하여 향후 일정 시간동안의 최근접 리스트를 출력할 수 있는 기법을 제안한다. 그룹 기반의 최근접 질의가 단일점 기반의 최근접 질의처리였다면 제안하는 기법은 단일점의 이동 패턴을 고려하여 선형 방정식을 추출한다. 추출한 선형 방정식은 주어진 벡터와 함께 움직이는 질의점의 방향을 포함한다. 이러한 알고리즘은 연속적인 최근접 질의(Continuous Nearest Neighbor, CNN) [6, 7]에 적용하여 효율적인 질의 처리 기법을 제안하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문에서 제안하는 기법에 대해 설명하고 3장에서 본 논문에서 제안하는 기법에 대해 실험을 수행하였으며, 4장에서 결론을 내려한다.

II. 제안하는 기법

이 장에서는 제안하는 기법인 움직이는 차량들간의 연속적인 최근접 질의 처리 기법에 대해 상세히 서술하려 한다. 우선 첫 절에 전반적인 기술에 대해 살펴보면, 두 번째 절에서 제안하는 기법의 1단계 기법인 센트로이드 계산 방법을 제안하고 마지막 절에서 센트로이드선을 기반으로 최근접을 찾는 방법에 대해 알아본다.

1. 움직이는 차량간 최근접 질의 처리 기법

각 객체의 집합이 R-tree에 의해 색인화되어 있고 질의점들의 집합 Q가 있다고 가정하면 제안하는 기법, DNN_MV (Dynamic Nearest Neighbor for Multiple Vehicles)를 위한 알고리즘은 다음과 같은 과정을 가

진다. 우선, 질의점들의 센트로이드 점을 찾는다. 센트로이드는 무게 중심으로서 모든 질의점들의 x, y 좌표를 각각 더한 다음 전체 질의점의 개수로 나누면 계산이 가능하다 [8, 9]. 다음으로, 센트로이드 점의 선형 특성을 가진 선형 방정식을 구한다. 마지막으로 구한 센트로이드 선을 바탕으로 DNN_MV를 구하는 과정을 가진다.

전통적인 그룹기반의 최근접 질의[5]는 입력값으로 최근접점을 계산한 정적 그룹점을 요구한다. 그러나 질의점이 주어진 벡터와 함께 이동할 때 전통적인 그룹기반의 최근접 질의는 지속적으로 이동하는 그룹의 최근접 질의에는 적합하지 않는다. 질의점들이 주어진 벡터를 가지고 지속적으로 이동할 경우에는 이러한 그룹기반의 최근접 질의는 심각한 탐색 오버헤드를 발생시킨다. 질의점들이 움직인다는 것은 최근접 또한 지속적으로 움직인다는 것을 의미한다. 이는 끊임없이 이동하는 질의점에 대한 최근접 질의를 처리하기 위해 끊임없이 탐색해야 하므로 탐색 비용이 많이 소요된다. 따라서 지속적으로 이동하는 차량인 질의점들에 대한 그룹기반의 최근접 질의를 처리하기 위한 효율적인 방법론에 대해 제안한다.

본 논문에서는 각각의 차량에는 GPS가 설치되어 있고 이는 주기적으로 자신의 위치를 중앙 서버에 업데이트하도록 가정한다. 다차원의 공간에서 데이터의 집합은 $P = \{p_1, \dots, p_N\}$ 이고 차량의 현재 위치의 집합은 $Q = \{q_1, \dots, q_M\}$ 이다. 결과값은 Q 에 포함된 모든 질의점들과 가장 가까운 거리를 가진 k 개의 데이터들을 포함한다. p 와 Q 와의 거리는 $dist(p, Q) = \sum_{i=1}^n |pq_i|$ 이다. 이때 $|pq_i|$ 는 데이터 위치 p 와 하나의 질의점 q_i 와의 유클리디안 거리이다. 본 질의는 차량위치의 집합 $Q = \{q_1, \dots, q_M\}$ 이고 결과는 모든 차량에 이동거리의 최소값을 반환한다.

2. 센트로이드 계산

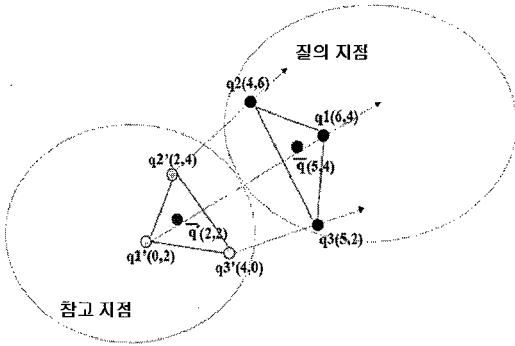
움직이는 차량의 그룹 기반의 최근접 질의를 처리하기 위한 일반적인 방법은 연속적 질의처리 방법

(Continuous Query Processing, CNN) [6, 7]을 사용한다. 하지만 이 방법은 많은 탐색 비용이 요구되므로 효율성을 높이기 위해 제안하는 기법 DNN_MV에서는 우선 차량의 위치 집합 Q 에서 센트로이드 q 를 계산한다. q 는 모든 차량으로의 거리를 모두 더한 값의 최소값이 되는 위치로서 $dist(q, Q)$ 가 최소가 되는 값이다. 기존의 논문[6]에서 제안한 기법은 질의 점의 벡터를 표현하는데 맞지 않기 때문에 하나의 점을 선택하며, 이를 센트로이드로 표현한다. 이를 위해서는 q 를 먼저 계산하고 q 주위의 P 집합 내의 데이터와의 거리를 계산하여 최근접 점을 찾아낸다.

센트로이드 q 의 좌표를 (x, y) 라고 할 때 (x_i, y_i) 는 $q_i \in Q$ 인 차량의 위치 좌표이다. 센트로이드 q 는 $dist(q, Q) = \sum_{q_i \in Q} |qq_i|$ 인 거리 측정 함수를 최소화하는 값을 가진다. 독립변수 x 와 y 와 관련한 부분 추출함수 $dist(q, Q)$ 가 센트로이드 q 에서 0이면, 아래 함수를 유추할 수 있다.

$$\begin{cases} \frac{\delta dist(q, Q)}{\delta x} = \sum_{i=1}^n \frac{x - x_i}{\sqrt{(x - x_i)^2 + (y - y_i)^2}} = 0 \\ \frac{\delta dist(q, Q)}{\delta y} = \sum_{i=1}^n \frac{y - y_i}{\sqrt{(x - x_i)^2 + (y - y_i)^2}} = 0 \end{cases} \quad (1)$$

수식 (1)은 반드시 움직이는 차량의 그룹에 대한 그룹기반의 최근접 질의를 만족한다. 그러나 n 의 개수가 4보다 크거나 같게 되면, 수식 (1)을 만족하는 점을 찾는 방법은 없다 [8, 9]. 기존의 그룹기반의 최근접 질의 방법은 기울기 하강 방법(gradient decent method) [10]이다. 우리는 수식 (1)을 만족하는 점이 n 이 4보다 작거나 같은 경우에는 기하학적 센트로이드(\bar{q})와 동일하다. 그러므로 그룹 기반의 최근접 질의는 $n \leq 4$ 인 경우에 성립하며 기하학적 센트로이드는 수식 (1)을 만족하는 점이다. 위에서 표현한대로, 모든 차량과의 거리가 가장 짧은 점 q 를 찾기 위해서는 기하학적 센트로이드, \bar{q} 를 적용할 수 있으며 기하학적 센트로이드는 모든 질의점의 평균값이다. 기하학적 센트로이드 \bar{q} 는 아래와 같이 표시할 수 있다. 즉, $\bar{q}(x, y)$ 가 Q 의 기하학적 센트로이드라 한다



<그림 2> 센트로이드 점을 찾는 과정
<Fig. 2> Finding a centroid point

면, 각각의 좌표는 아래와 같다.

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i, \quad \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i \quad (2)$$

그룹 기반의 최근접은 센트로이드 점 \bar{q} 를 사용하여 효율적으로 탐색할 수 있다. 그러나 질의점의 총 수가 4보다 크다면 수학적 해결방법은 아직까지는 없다 [8]. 그러나 $n \leq 4$ 인 경우에는 사용하면 매우 효율적으로 움직이는 차량의 최근접에 대해 처리할 수 있다.

각 차량은 이동할 때 주기적으로 자신의 위치를 서버에 보고한다. 따라서 서버는 주기적으로 가 차량의 최근의 위치에 대해 파악하고 있다. 이러한 최근의 위치는 매우 동적으로 이동하는 차량의 그룹 기반의 최근접 질의를 처리하는데 참고할 수 있는 포인트를 제공한다. 그림 2는 질의과정에서 먼저 센트로이드 점을 계산하는 과정에 대해 보여준다. 차량은 위에서 언급한대로 주어진 벡터를 가지고 이동한다. 본 그림에서는 두 점들의 집합이 보이는데 하나는 질의가 발생한 시점의 각각의 차량 위치(참고지점)이고, 하나는 질의가 처리될 각각의 차량 위치(질의지점)이며, 질의지점은 각각의 차량의 벡터와 함께 계산이 가능하다. 질의지점의 차량 위치는 $Q = \{q_1, q_2, q_3\}$ 이고 참고지점의 질의점은 $Q' = \{q'_1, q'_2, q'_3\}$ 으로 표현하고 각각의 센트로이드는 \bar{q} 와 \bar{q}' 로 표시한다.

이 경우 기하학적 센트로이드는 선형의 특성을 가

지며 이는 아래와 같이 표현가능하다.

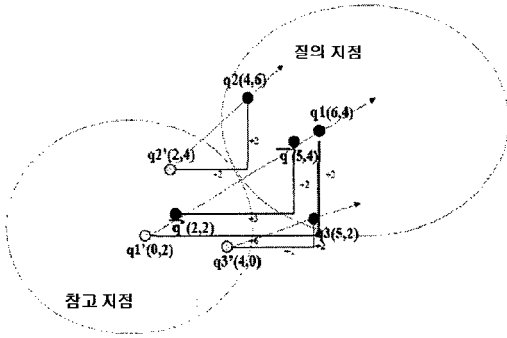
정리 1 : 임의의 n 점이 주어진 벡터를 가지고 이동할 때 n 점의 기하학적 센트로이드 점은 선형으로 이동한다.

증명 : 각 n 개의 질의점이 2차평면상에서 각각 $q_1(x_1, y_1), q_2(x_2, y_2), \dots, q_n(x_n, y_n)$ 이라고 가정한다. 기하학적 센트로이드 $\bar{q}(\bar{x}, \bar{y})$ 는 수식 (2)와 같이 계산한다. 각각의 점들은 자신의 벡터와 함께 이동한다. 질의점은 x 축으로 δ_i 만큼 이동하고 y 축으로 ϵ_i 만큼 이동한다. 그러므로 기하학적 센트로이드는 x 축으로 $\frac{1}{n} \sum_{i=1}^n \delta_i$ 만큼 이동하고 y 축으로 $\frac{1}{n} \sum_{i=1}^n \epsilon_i$ 만큼 이동한다. 따라서 기하학적 센트로이드 \bar{q} 는 각각의 질의점이 이동하는 평균값과 함께 선형으로 이동한다.

정리 1은 그림 2에서 설명이 된다. 각각 참고지점의 위치 집합 $Q' = \{q'_1, q'_2, q'_3\}$ 의 센트로이드 좌표는 (2,2)이고 질의지점의 위치 집합 $Q = \{q_1, q_2, q_3\}$ 의 센트로이드 좌표는 (5,4)이다. 이 두 점은 질의점의 평균값과 함께 이동한다. 다시 말하면, $Q' = \{q'_1, q'_2, q'_3\}$ 은 주어진 벡터를 가지고 이동하고 기하학적 센트로이드는 각 질의점의 평균값을 가지고 이동한다.

3. 센트로이드 선을 바탕으로 최근접 질의 처리

그림 3에서 보는 바와 같이 질의지점 $Q' = \{q'_1, q'_2, q'_3\}$ 은 $Q = \{q_1, q_2, q_3\}$ 쪽으로 이동한다. 기하학적 센트로이드는 (2,2)에서 (5,4)로 움직이며 질의점의 평균 이동 값은 각각 $\left(\frac{2+6+1}{3} = 3, \frac{2+2+2}{3} = 2\right)$ 이다. 그러므로 질의점이 주어진 벡터와 함께 이동할 때 기하학적 센트로이드는 각각의 질의 이동 값의 평균값과 함께 선형으로 이동한다. 이러한 특징을 이용하면 기하학적 센트로이드의 움직임은 선형방정식으로 표현된다. 다시 말하면, 선형방정식은 참고지점의 점들, $Q' = \{q'_1, q'_2, q'_3\}$ 와 질의지점의 점들, $Q = \{q_1, q_2, q_3\}$ 의 센트로이드를 연결하는 직선이 되며, 이 선형방정식은 다음과 같다.



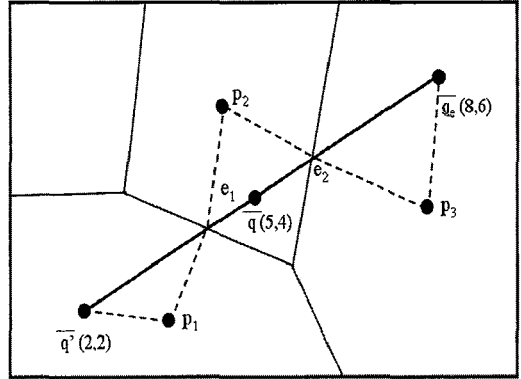
<그림 3> 센트로이드의 선형 특성

<Fig. 3> Linear property of centroid point

$$y - y_1 = \frac{y_2 - y_1}{x_2 - x_1} (x - x_1) \quad (3)$$

위 수식 (3)에 의하여 시작점이 (\bar{q}) 인 질의점의 선형방정식을 표현할 수 있다. 이러한 선형방정식에서 그룹기반의 최근접 질의를 수행하기 위해서 지속적인 최근접 탐색(CNN) [12]이 적용된다. 그림 4에서 보는 바와 같이 CNN의 경우에는 입력된 질의가 시작점이 (\bar{q}) 이고 종점이 (\bar{q}_e) 이라면, 이는 선 (\bar{q}) 과 (\bar{q}_e) 사이를 잇는 선에서 가장 가까운 최근접 탐색을 수행할 것이다. 시작점 (\bar{q}) 은 참고지점에서 기하학적 센트로이드이다. 위에서 설명한 바와 같이 센트로이드 점은 $\bar{q}(2,2)$ 와 $\bar{q}(5,4)$ 가 된다. 그리고 난후 선형방정식은 $y - 2 = \frac{4 - 2}{5 - 2} (x - 2)$ 로 구해지며, 정리하면 $y = \frac{2}{3}x + 2$ 가 된다. 이 방정식에 따라 직선의 종점 (\bar{q}_e) 이 정해진다. 종점 (\bar{q}_e) 은 각 어플리케이션에 따라 달라지므로 동적으로 결정할 수 있다. 제안한 방법에서 종점 (\bar{q}_e) 은 기본적으로 \bar{q} 와 \bar{q} 와의 거리의 두 배로 결정한다. 이 방법으로 질의 선을 사용해서 움직이는 차량에서 그룹기반의 최근접 질의를 수행할 수 있다.

제안하는 DNN_MV 질의의 목적은 그룹기반의 최근접을 찾는 것이다. DNN_MV질의는 선분인 $L(\bar{q}, \bar{q}_e)$ 으로 표현할 수 있다. 그림 4에서는 질의 선분 (\bar{q}, \bar{q}_e) 으로부터 그룹기반의 최근접을 찾는 과정을 보여준다. 질의의 결과값은 선분 (\bar{q}, \bar{q}_e) 상에서 세 개



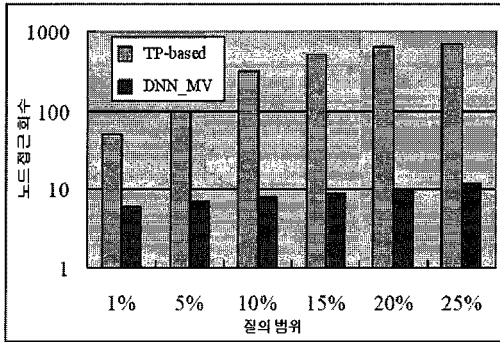
<그림 4> CNN기반의 DNN_MV 탐색

<Fig. 4> CNN-based search for DNN_MV

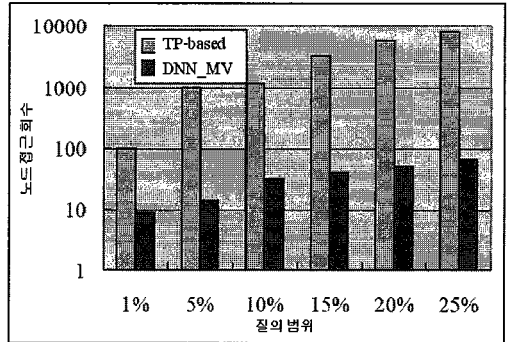
의 데이터인 (p_1, p_2, p_3) 이 된다. p_1 의 경우에 선분 (\bar{q}, e_1) 상에서 최근접 결과값이 되고, p_2 는 선분 (e_1, e_2) 에서 최근접 결과값이 된다. 비슷하게 p_3 는 선분 (e_2, \bar{q}_e) 에서 최근접이 된다. 이와 같은 방법으로 주어진 선분으로 질의값은 $(\bar{q}, e_1), p_1, (e_1, e_2), p_2, (e_2, \bar{q}_e), p_3$ 이 된다. 본 논문에서 제안한 방법은 CNN 탐색 방법과 비슷하다. 그러나 제안한 방법은 지속적으로 움직이는 차량에서 그룹기반의 최근접을 탐색하여 기존의 CNN에서는 수행하지 못한 질의를 처리한다.

III. 검증 실험

제안한 기법의 효율성을 실험하기 위해 본 논문에서는 두 개의 실제 데이터셋을 사용하였다. 하나는 북아메리카 지역의 24,493의 유명한 지점(POI)을 가진 데이터셋(CA Dataset) [11]이고, 다른 하나는 미국의 아이오와, 캔자스, 미주리, 네브라스카 지역의 194,971 지점(POI)을 가진 데이터 셋(ST Dataset) [11]이다. 각각의 데이터셋의 POI는 x,y좌표로 구성되어 있으며, 하나의 데이터는 4bytes의 크기를 가진다. 모든 실험에서는 Pentium 3.2GHz CPU와 1GByte 메모리를 가진 PC에서 실험하였다. 색인으로 사용한 R*-tree [12, 13]의 페이지 사이즈는 1KByte로 정하였고 노드당 50개의 엔트리가 들어가도록 하였다. 차량은 약 200그룹으로 정하였고, 질의는 전 맵에 걸쳐 균등하게 발생하도록 하였다. 차량의 이동방향은



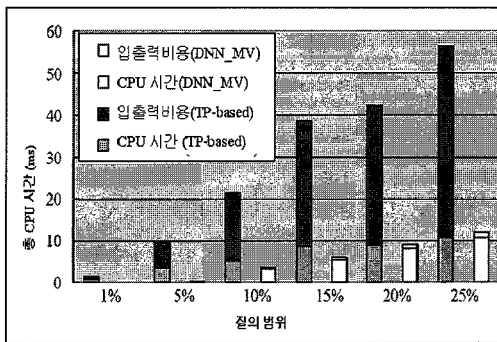
(a) CA Dataset



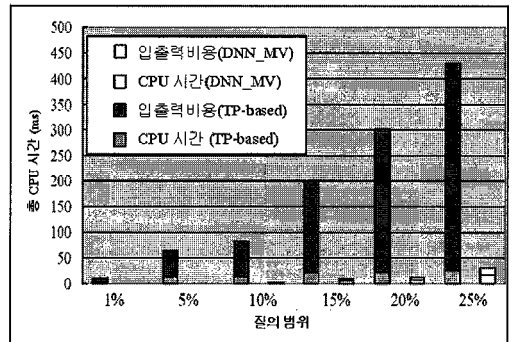
(b) ST Dataset

<그림 5> 질의 범위 vs 노드 접근 회수

<Fig. 5> Query length vs Node accesses



(a) CA Dataset



(b) ST Dataset

<그림 6> 질의 범위 vs 총 CPU 시간

<Fig. 6> Query length vs Total CPU time

(0, 2π)에 걸쳐 균등하게 분포시켰으며, 모든 질의는 균등한 시간에 걸쳐서 발생하도록 하였다. 질의의 길이는 어플리케이션마다 다른 점을 감안하여 다양한 길이를 적용시켜 길이가 1%에서 25%에 걸쳐서 분포시켰다.

센트로이드의 선형 특성에 따라서 두 개의 연속적 질의 처리 알고리즘을 비교하였다. 제안한 알고리즘 DNN_MV와 일정한 시간동안 반복해서 최근접 질의를 처리하는 TP-based (Time parameterization-based) 기법 [14]을 비교하였다. 그림 5는 이 두 방법을 비교 분석한 그래프이다. 각각 두 개의 데이터셋에 대해 비교 분석하였으며, 질의의 범위에 따라 데이터 노드 접근 회수가 어떻게 되는지 분석하였다. 그래

프에서 보는 바와 같이 제한한 DNN_MV는 TP-based 기법에 비해 훨씬 적은 노드를 접근함으로써 질의 처리 시간을 줄임을 알 수 있다. 이는 질의 범위가 길수록 더욱 차이가 많이 벌어짐을 보여준다.

그림 6은 질의에 따라 서버에서 처리하는 시간을 측정된 결과이다. 두 개의 데이터셋 모두에서 제안한 기법이 기존 기법에 비해 입출력 시간과 CPU 시간이 포함된 시간에서 훨씬 적게 드는 것을 볼 수 있다. 본 실험에서 보는 바와 같이 CPU 시간은 두 방법 모두 질의 범위가 길어짐에 따라서 증가함을 보인다. DNN_MV의 경우 질의 크기가 증가하면 총 선분 중 나뉘는 구간이 많아져 더 많은 거리 계산이 필요하다. TP-based의 경우에는 버퍼가 입출력 비용

의 대부분을 흡수한다. 그러므로 CPU시간의 백분율은 질의 범위가 증가함에 따라 입출력 비용을 넘어 서게 된다.

이들 실험의 결과 제안한 DNN_MV는 모든 경우에서 기존의 TP_based 기법보다 좋은 성능을 보인다. 이는 제안한 기법이 단지 모든 분기점에서 데이터 셋을 한번만 검색하기 때문이고 TP_based의 경우 입력 선분 길이가 매우 짧을때만 더 좋은 성능을 보인다.

IV. 결론 및 향후 연구

본 논문에서는 여러 대의 차량이 짧은 거리를 이동해 빠르게 모이게 하기 위한 방법론을 제안하였다. 기존 연구는 이동하지 않는 객체들을 다루고 있어 움직이는 차량에 적용하기엔 어려운 점이 있어 본 논문에서는 이동하는 차량들에게 효율적인 차량간 최근접 질의 처리 기법을 제안하였다. 본 기법은 각 차량의 움직이는 방향과 속도를 기반으로 모든 차량이 최소 시간에 모일 수 있는 최근접 질의점을 찾을 수 있다. 본 기법은 그룹 내에서 센트로이드를 찾은 후 이를 하나의 선분으로 표시하여 연속된 최근접 질의를 수행함으로써 질의를 처리하였다. 실험 결과는 기본기법에 비해 제안하는 기법이 움직이는 차량의 최근접 질의 처리에 효율적임을 보여준다.

본 연구를 통해 많은 차량용 어플리케이션에 응용할 수 있다. 친구로 등록된 차량끼리 원하는 시간에 만날 수 있는 장소를 바로 제공할 수 있으며, 특히 서로의 향후 스케줄을 고려하여 약속장소를 정함으로써 사용자의 만족도가 상승할 수 있다. 본 연구는 위치기반서비스의 가장 많이 사용하는 최근접 질의 중 각기 다른 속도를 가진 그룹기반의 최근접 질의를 연구함으로써 특히, 속도의 편차가 큰 차량에서 그 효율성이 극대화된다.

향후 연구로서 차량이 직선의 벡터값을 가지고 직선으로 이동하는 것이 아닌 다양한 도로위의 길로 이동할 때 처리할 수 있는 최근접 질의를 수행할 수 있는 질의 처리기법에 대해 향후 연구 주제로 삼고자한다.

참고 문헌

- [1] G. Kollios, D. Gunopulos, and V. J. Tsotras. Nearest Neighbor Queries in Mobile Environment. *International Workshop on Spatio-Temporal Database Management*, pages 119-134, July 1999.
- [2] D. Papadias, Y. Tao, K. Mouraridis and C. K. Hui. Aggregate Nearest Neighbor Queries in Spatial Databases. *ACM Transactions on Database Systems*, 30(2):529-576, June 2005.
- [3] N. Roussopoulos, S. Kelly, and F. Vincent. Nearest neighbor query. *In Proceedings of ACM SIGMOD International Conference on Management of Data*, pages 71-79, June 1995.
- [4] Z. Song and N. Roussopoulos. K-Nearest Neighbor Search for Moving Query Point. *In Proceedings of the 7th International Symposium on Spatial and Temporal Databases*, pages 79-96, July 2001.
- [5] D. Papadias, Q. Shen, Y. Tao, and K. Mouratidis. Group Nearest Neighbor Queries. *In Proceedings of the 20th International Conference on Data Engineering*, pages 301-313, March 2004.
- [6] S. Saltenis, C. Jensen, S. Leutenegger, and M. Lopez. Indexing the Positions of Continuously Moving Objects. *In Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 331-342, June 2000.
- [7] Y. Tao, D. Papadias, and Q. Shen. Continuous Nearest Neighbor Search. *In Proceedings of the 28th International Conference on Very Large Data Bases*, pages 287-298, Aug. 2002.
- [8] T. M. Apostol and M. A. Mnatsakanian. Finding Centroids the Easy Way. *Math Horizons*, pages 7-12, 2000.
- [9] Geometric Centroid. available at [http://www.math-world.wolfram.com/Geometric Centroid.html](http://www.math-world.wolfram.com/Geometric%20Centroid.html).
- [10] S. Hochreiter, A. S. Younger, and P. R. Conwell. Learning to Learn Using Gradient Descent. *In Proceedings International Conference on Artificial*

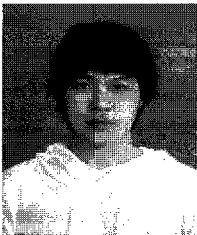
- Neural Networks*, pages 87-94, Aug. 2001
- [11] Spatial Datasets. available at <http://www.maproom.psu.edu/dcw>.
- [12] N. Beckmann, H. P. Kriegel, R. Schneider, and B. Seeger. The R*-tree: An Efficient and Robust Access Method for Points and Rectangles. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 322-331, May 1990.
- [13] A. Guttman. R-Trees: A Dynamic Index Structure for Spatial Searching. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 47-54, June 1984.
- [14] Y. Tao and D. Papadias. Time-Parameterized Queries in Spatio-Temporal Databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 334-345, June 2002.

저작소개



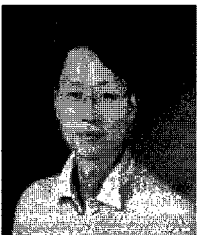
이 명 수 (Lee, Myong-Soo)

2007년 3월 ~ 현재 : 고려대학교 컴퓨터통신공학부 박사과정
2004년 3월 ~ 2006년 2월 : 고려대학교 컴퓨터통신공학부 석사
1997년 3월 ~ 2004년 2월 : 고려대학교 컴퓨터통신공학부 학사



심 규 선 (Shim, Kyu-Sun)

2009년 3월 ~ 현재 : 고려대학교 컴퓨터통신공학부 석사과정
2003년 3월 ~ 2009년 2월 : 고려대학교 컴퓨터통신공학부 학사



이 상 근 (Lee, SangKeun)

2003년 ~ 현재 : 고려대학교 컴퓨터통신공학부 부교수
2000년 ~ 2001년 : University of Tokyo 특별 방문 연구원
1996년 3월 ~ 1999년 2월 : 고려대학교 컴퓨터학과 박사
1994년 3월 ~ 1996년 2월 : 고려대학교 컴퓨터학과 석사
1990년 3월 ~ 1994년 2월 : 고려대학교 컴퓨터학과 학사