이동객체의 궤적에 대한 연속 최근접 질의에 관한 연구*

Study on Continuous Nearest Neighbor Query on Trajectory of Moving Objects

정지문**

목 차

Abstract

Researches for NN(nearest neighbor) query which is often used in LBS system, have been worked. However, Conventional NN query processing techniques are usually meaningless in moving object management system for LBS since their results may be invalidated as soon as the query and data objects move.

To solve these problems, in this paper we propose a new nearest neighbor query processing technique, called CTNN, which is possible to meet continuous trajectory nearest neighbor query processing. The proposed technique consists of Approximate CTNN technique which has quick response time, and Exact CTNN technique which makes it possible to search accurately nearest neighbor objects. Experimental results using GSTD datasets showed that the Exact CTNN technique has high accuracy, but has a little low performnce for response time. They also showed that the Appriximate CTNN techinque has low accuracy comparing with the Exact CTNN, but has high response time.

* 이 논문은 2005년 남서울대학교 학술연구조성비 지원에 의하여 연구되었음.
** 남서울대학교 컴퓨터학과 교수, jmchung@nsu.ac.kr.

Ⅰ. 서론

최근 무선 인터넷 및 모바일 컴퓨팅 기술이 급속히 발전함에 따라 위치 기반 서비스 관련 기술 개발이 활발히 진행되고 있다. 이 서비스들은 실세계 시공간 데이터에 다양한 질의 처리 기법이나 효율적인 데이터 저장 및 관리 기법들을 적용하여 사용자에게 원하는 정보를 제공하고자 한다. 우리가 자주 사용하는 질의 처리 기법 중 하나인 최근접 질의는 사용자의 선택 위치와 가장 가까운 곳에 존재하는 객체를 결과로 반환하며, 위치 기반 서비스 내에서 다음의 예제와 같이 적용될 수 있다: A씨는 집에서 핸드폰으로 가장 가까운 곳에 위치한 피자집을 검색하여 피자를 주문한다(정적-정적). 운전 중인 B씨는 자신의 차에 장착된 네비게이터(navigator)를 통해서 가장 가까운 곳에 위치한 주유소를 검색한다(동적-정적). 항해 중인 선박 C는 근처에서 항해하고 있는 주유선을 검색하여 급유를 요청하고자 한다(동적-동적). 고속버스 운전기사 D씨는 고속도로 위에서 사고를 겪었다. D씨는 가장 먼저 자신의 위치에 도착 가능한 버스를 검색하여 도움을 요청하였고, 승객들은 모두 다른 버스를 이용하여 목적지에 무사히 도달할 수 있었다(정적-동적). 이 때, 괄호 안의 요소는 질의-데이터 객체 타입을 나타낸다.

이와 같이 위치 기반 서비스들은 여러 타입의 데이터 객체들에 대한 질의를 처리하며, 효율적인 성능을 위해 이에 따른 여러 기법들이 제안되었다. 공간 객체와는 달리, 이동객체들은 시간의 흐름에 따라 연속적으로 자신의 위치를 변경하며 움직이기 때문에 어떤 한 시점에서 계산된 최근접 질의 결과는 다른 시간에서 유효하지 않을 수 있다. 하지만 대부분의 시스템이 사용하는 최근접 질의 처리들은 이러한 속성을 무시한 채, 이동객체 궤적의 시작 시점에서만 객체간 거리를 비교하여 최근접 객체를 선택하거나, 샘플링 된 일정 간격 시간점 위에서 거리를 비교하여 결과를 반환한다. 따라서 전체 질의 시간에 대해 항상 유효한 정보를 사용자에게 제공할 수 없다. 이를 해결하기 위해서 연속 또는 영속 질의 처리 개념[1]을 도입한 기법들이 제안되기도 하였지만, 이들 역시 객체 궤적 전체에 대한 최근접 질의를 행하지 않기 때문에 정확한 결과를 제공하기에는 미흡하다. 즉 이들 기법은 서로 가까워지는지 멀어지는지 등, 궤적 정보를 전혀 고려하지 않기 때문에 부정확한 결과를 보이며, 또한 하나의 최근접 객체를 선택해야 하는 질의에서 여러 후보 객체를 가지는 경우, 결과 선택 기준이 명확하지 않기 때문에 임의로 결과를 선택하게 되며 이는 부정확한 결과를 낳을 수 있다.

따라서 이 논문에서는 기존 연구의 문제점을 해결하기 위하여, 객체간 궤적 정보를 비교하면서 연속적으로 질의와 가장 가까운 거리를 유지하며 움직이는 객체를 찾는 것이 가능한 새로운 최근접 질의 처리 기법인 연속 궤적 최근접(CTNN) 질의 처리 기법을 제안하였다. 이 기법은 궤적 정보 비교를 통해 질의와 가장 가까운 위치를 유지하며 움직이는 객체를 정확히 탐색할 뿐만 아니라 여러 후보 객체가 있을 때 k값(1또는 n)에 따라 결과를 선택할 수 있는 기준을 제공하며, 질의 객체의 궤적 상에서 최근접 객체 정보가 변경되는 시점을 연속적으로 지정한 후 그 시간점 위에서 결과가 변경되는지 여부를 비교함으로써 질의 전체에 대해 항상 유효한 결과를 얻을 수 있도록 하였다. 또한, 이 논문에서는 응답 시간을 빠르게 하기 위한 Approximate CTNN 기법과 정확한 최근접 객체를 탐색하기 위한 Exact CTNN 기법을 제안하였다.

논문의 나머지는 다음과 같이 구성되어 있다. 2

절에서는 기존 연구들을 살펴보고, 3절에서는 이 논문에 적용된 기본 모델과 가정들을 살펴보며, 4절에서는 이 논문에서 제안하는 CTNN 기법들에 대한 질의 처리 과정과 알고리즘을 소개하고, 5절에서는 제안 기법의 성능평가를 보인다. 그리고 마지막으로 6절에서 결론 및 향후 연구를 제시한다.

Ⅱ. 관련 연구

최근접 질의 처리 기법들은 멀티미디어 데이터베이스[2], 데이터 마이닝[3], 공간·시공간·이동체 데이터 베이스[4, 5, 6, 7, 8] 등 광범위한 분야에서 연구되고 있으며, 최근접 질의 처리를 위한 인덱스나[9, 10], 응답 시간을 줄이기 위한 근사(approximate) 질의 처리 기법[11, 12, 13, 14]이 제안되기도 하였다. 이 절에서는 질의 객체와 데이터 객체가 모두 이동객체인 경우에 적용 가능한 최근접 질의 처리 기법들을 살펴본다.

[15]는 쌍대성 변환(Duality Transform) 기법을 사용하여 (x, y) 평면 상의 이동객체 궤적 선분들을 (v, a) 평면 (속도, y절편) 상의 점으로 변환하고, 점 간의 유클리안 거리를 계산하여 최근접 객체를 선택한다. 이는 모든 질의-데이터 객체 타입에 적용 가능하지만 연속적인 최근접 탐색이 불가능하며 3차원 (x, y, t) 공간 상의 선분으로 표현되는 이동객체 궤적에는 부적합하다.

[16]은 TPR-tree[17]을 사용하여 데이터를 인덱스하고, DF 탐색 방법[4]에 따라 미분 함수를 사용하여 노드 사각형과 질의점 사이의 거리를 계산한 후 [now, now+△] 간격마다 가장 가까운 거리를 가지는 객체를 찾아 결과로 반환하는, 영속 최근접 질의 처리 기법을 제안하였다. 이는 단순 탐색 기법을 따르기 때문에 한 데이터에 대해 여러 번 중복된 거리 계산을 행하기도 하고, 이전에 탐색된 최근접 객체가 현재에도 결과로서 유지되는지 아닌지를 비교하는 작업들을 자주 요구한다. 또한 거리 계산 함수는 객체의 위치나 속도에 의존적이기 때문에 이 값들이 변경될 때마다 함수가 재계산되므로 큰 오버헤드를 초래할 수 있다.

[18]은 TPNN(time-parameterized)질의와 CNN(continuous)질의를 제안하였다. 이들은 최근접 객체를 선택하면서, 전체 질의 시간 선분 위에서 최근접 객체 정보가 변경될지도 모르는 (TPNN) 또는 변경되는 (CNN) 시간점을 계산하고, 각각의 최근접 객체 정보와 유효시간 인터벌을 시간 리스트 TL에 저장하였다가 질의 마지막 시간에 결과로 반환하는 연속 최근접 질의이다. 즉, 결과는 TL={⟨a, [t1, t2]⟩, ⟨b, [t2, t3]⟩}과 같이 여러 개의 연속적이고 분리된 하위 시간 인터벌과 각 인터벌 내에서 유효한 최근접 객체 정보로 구성된다. TPNN질의는 모든 객체에 대해 질의와 가장 가까워지는 가까운 미래 시간을 미리 계산하고, 계산된 시간 값에 따라 순서대로 계산된 시간 점 위에서 실제 최근접 객체가 되는지를 검사한 후, 결과를 선택한다. TPNN 질의의 계산적 오버헤드를 줄이기 위하여 제안된 것이 CNN 질의이며, 이는 질의 시작 시간에서 객체들과 질의 q 사이의 거리를 계산한 후 거리를 오름차순으로 정렬하고 첫 번째 최근접 객체 p를 선택한다. 다음으로 다른 객체가 p와 q 사이에 생성된 근접원(vicinity) 안에 포함되는지를 거리 순서대로 비교하고, 포함된다면 그 객체를 최근접 객체 p'로 선택하면서 유효시간 인터벌을 계산한다. 그리고 p'와 q 사이에 또 근접원을 그리고, 동일 과정을 질의 마지막 시간까지 반복한 후 TL내의 결과를 반환한다. 질의와 데이

터가 모두 이동객체인 경우를 위한 CkNN 질의는 최근접 객체가 갱신되는 시점의 타임스탬프 집합으로 구성된 분할 리스트를 유지하여, 분할 리스트에 유지되는 타임스탬프 상에서의 kNN 객체를 탐색하게 되며, 따라서 다른 타임스탬프는 고려할 필요가 없게 된다. 그러나, CkNN 기법의 분할 리스트를 결정하기 위해서는 최근접 객체가 갱신되는 순간을 다루기 위해 모든 시간에서의 최근접 객체에 대한 갱신여부를 체크하게 된다. 또한, CkNN 기법을 이동객체의 궤적에 적용할 경우 타임스탬프 단위에 따라 이동객체의 위치를 추정하여 최근접 객체 여부를 비교하게 되므로, 타임스탬프 단위에 따라, 예를 들면 1분, 10분, 1시간, 서로 다른 객체가 최근접 객체로 선택되게 되며, 따라서, 부정확한 결과값을 반환할 수 있다.

Ⅲ. 질의 모델

이 절에서는 CTNN 질의의 기본 데이터 구조와 질의 타입을 설명하며, 이 논문이 포함하고 있는 몇 가지 가정 및 표기들을 소개한다.

3.1. 데이터 모델

이 논문은 시간의 흐름에 따라 자신의 위치를 변경하며 연속적으로 움직이는 이동 점 객체 예를 들면, 자동차, 사람, 비행기, 선박 등에 대한 최근접 질의를 대상으로 한다. 객체의 위치 정보는 GPS 등의 장치를 통하여 주기적으로 추출되고, 객체의 삽입, 삭제, 속도나 이동 방향의 변경 등과 같은 갱신이 발생할 때 데이터베이스에 저장된다. 즉, 이동객체는 갱신이 발생되는 각 시간 t에서 〈idi, (x, y), t〉 형태로 저장되며, 이 때 idi는 객체 id의 i번째 저장 정보를, (x, y)는 시간 t에서의 공간 좌표를 나타내고, 속도나 방향 정보는 저장하지 않는다.

[정의 1] 궤적 Tri는 이동객체 i가 움직인 경로이며, 3차원 공간 (x, y, t) 상의 폴리라인(polyline) 즉, 선분 세그먼트 S들의 집합으로 표현된다. 즉 n개의 선분 세그먼트로 구성된 이동객체 i의 궤적 Tri는 다음과 같다. 여기서, Sin은 i번째 객체의 n번째 세그먼트를 의미한다.

$$Tri = \{Si1\ Si2, Si3, \cdots, Sin\}$$

[정의 2] 이동객체의 궤적 Tri을 구성하는 선분 세그먼트 Sin는 데이터베이스 내에서 연속되어 나타나는 두개의 데이터 점으로 구성된다. 여기서 idn은 객체 i의 식별자는 id이고, n번째 갱신되었음을 나타내며, xn, yn, tn은 각각 n번째 공간 좌표와 시간을 나타낸다.

$$Sin = \{\langle idn, (xn, yn), tn\rangle, \langle idn+1, (xn+1, yn+1), tn+1\rangle\}$$

이동객체 궤적은 크게 두 가지, 자유궤적과 제약궤적으로 구분된다[19]. 제약궤적은 자유궤적의 특별한 경우로 철로 위의 기차, 고속도로 위의 자동차 등과 같이 객체의 이동 경로가 제한된 것을 말한다. 우리는 미리 이동 경로를 예측할 수 있는 객체를 '제약궤적을 가진 객체'로 간주하며, 이 논문에서 제안되는 CTNN 기법들은 버스, 비행기, 기차 등 제약궤적 이동 점 객체에 적용 가능하다.

데이터베이스에 저장되지 않은 시점에서의 공간 좌표는 다음의 시간 종속적 선형 위치 추정 함수를

사용하여 추정될 수 있다.

$$f(t)=\left(\frac{x_{i+1}-x_i}{t_{i+1}-t_i}(t-t_i)+x_i\ ,\ \frac{y_{i+1}-y_i}{t_{i+1}-t_i}(t-t_i)+y_i\right) \quad (\text{ti} < \text{t} < \text{ti+1})$$

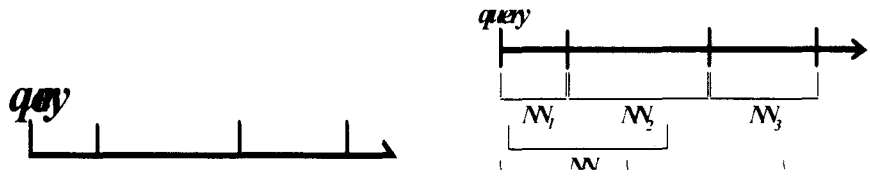
이 함수는 데이터베이스에 저장된 시간 ti와 ti+1에서의 좌표 값을 이용하여, 그 사이에 존재하는 시간 t에서의 객체 위치를 계산하기 때문에, 한 선분 세그먼트의 양 끝점 정보만으로 한 선분 세그먼트 위의 객체 공간 점 위치를 추정할 수 있다.

3.2. 질의 타입

범위(range) 질의는 다른 타입의 질의를 처리하는 과정에서 데이터의 양을 줄이는 전처리 작업으로 많이 사용되며, CTNN 질의 역시 시간 범위 질의를 사용하여 검색 대상이 되는 데이터 객체들을 필터링 한다. [20]에 따르면 시공간 최근접 질의는 크게 두 가지, 시간 범위 질의와 공간 최근접 질의의 합, 공간 범위 질의와 시간 최근접 질의의 합으로 구분된다. CTNN은 기본적으로 전자의 경우를 따르며, 좀 더 엄밀히 말해서 최근접 객체가 선택된 한 시점으로부터 결과가 변경되는 가장 가까운 미래 시간을 빠르게 찾아내고 그 시간점에서 공간적으로 가장 가까운 위치를 가지는 객체를 선택하기 때문에 공간 최근접 탐색과 시간 최근접 탐색이 동시에 가능한 시공간 최근접 질의 타입이다.

또한 전체 질의 시간 상에서 최근접 객체가 변경될 시점을 선택한 후 그 시간점에서 실제 변경 여부를 판단하여 저장하기 때문에 항상 유효하고 연속적인 결과를 얻을 수 있다. 즉, 연속 최근접 질의 타입을 가진다.

그리고 CTNN 질의는 사용자가 몇 개의 최근접 객체를 결과로 원하는지에 상관없이 조건을 만족시킬 수 있다는 장점을 가진다. 즉, 최근접 질의, k-최근접 질의가 모두 가능하며, k값은 사용자의 지정에 따르거나 결과로 나타날 수 있는 모든 후보 객체 개수가 될 수 있다. 그림 1은 최근접 질의와 k-최근접 질의의 결과 객체 정보와 유효시간 인터벌간 차이를 보인다. 예를 들면, 질의 시간이 [ts , te]로 주어졌을 때, 그림 1(a)의 최근접 질의 결과는 TL = {〈NN1 , [ts , t1]〉, 〈NN2 , [t1 , t2]〉, 〈NN3 , [t2 , te]〉} 형태가 된다. 즉, 전체 질의 시간 상에서 각각의 개별적이고 분리된, 연속적인 유효시간 인터벌과 각 인터벌 내에서 유효한 하나의 최근접 객체 정보로 구성된다. 또한 그림 1(b)의 k-최근접 질의 결과는 TL = {〈NN1 , [ts , t3]〉, 〈NN4 , [t1 , t5]〉, 〈NN5 , [ts , t4]〉, 〈NN2 , [t3 , t6]〉, 〈NN3 , [t6 , te]〉} 형태가 되며, 이는 시간 인터벌이 서로 겹치는 여러 하위 유효시간 인터벌과 해당 인터벌 내에서 유효한 최근접 객체들을 포함하게 된다.



(a) 최근접 질의 (b) k-최근접 질의

그림1. CTNN 질의에서의 결과 유효시간 인터벌

3.3. 가정 및 표기

CTNN 질의는 다음의 가정을 바탕으로 처리된다.

1. 어떤 객체 p가 최근접 객체로 선택된 후 1분 이내에 변경된다면, p는 무시되고 유효시간 인터벌은 이전 또는 이후 인터벌에 포함된다. 이 시간 인터벌은 하나의 결과를 표현하기에는 짧은 시간이라고 판단되며, 실 세계에서는 무의미하게 사용될 수 있다. 그렇지만, 응용에 따라 시간 단위를 변경할 수 있다.

2. 두 객체 사이의 거리가 1m 이내인 경우, 이 객체들은 서로 만나거나 교차한다고 간주한다. 실 세계에서 객체들이 같은 시간에 같은 좌표를 가지는 경우는 충돌을 의미하며, 이러한 상황은 거의 발생하지 않는다. 따라서 어떤 시간 점 위에서 1m 이내에 존재하는 두 객체는 같은 시간에 동일 위치에 존재하는 객체로 간주한다.

그리고 다음의 표기들을 사용한다.

Ⅳ. 연속 궤적 최근접 질의

이 절에서는 CTNN 질의에 사용되는 기본 알고리즘과 계산 과정들을 소개한 후, 이 논문이 제안하는 두 가지 질의 처리 기법, Approximate와 Exact CTNN의 알고리즘을 소개한다. 그리고 k-CTNN 질의로의 확장을 설명한다.

4.1. CTNN 질의 기본 알고리즘

CTNN 질의는 먼저, 주어진 질의 시간 범위 내에 존재하는 모든 살아 있는 데이터 객체들을 검색하고 질의 시간 인터벌로 이 세그먼트들의 시간 인터벌을 Clipping한 후, 최근접 질의 처리를 통해 질의 궤적과 가장 가까운 거리를 유지하는 세그먼트를 결과로 선택한다. 최근접 객체를 선택하는 과

표1. 표기

□ai	객체 a의 i번째 세그먼트 (i ≥1 정수)
(xat, yat)	시간 t에서 객체 a의 공간 점 좌표
(xai, yai)	객체 a의 i번째 저장 좌표
Δx (Δy)	x(y) 축에서 객체가 단위 시간당 움직인 거리
xab (yab)	객체 a, b의 x(y) 좌표 차이 값
|a - b|t	시간 t에서 a, b사이의 공간 거리 = √(xabt)2 + (yabt)2
/p	세그먼트 p의 기울기 = Δx/ Δy
∂p	세그먼트 p의 변위 값. 객체가 단위 시간당 움직인 거리

정은 크게, 연속 질의를 위한 평가시간 설정 단계와 궤적 질의를 위한 객체 궤적 정보 비교 단계로 구성된다. 평가시간이란, 질의 객체의 전체 궤적 선분 상에서 최근접 객체가 획득되는 시간, 또는 최근접 객체 정보가 변경되는 시간을 말한다. 즉, 질의가 평가되는 시간이며, 탐색된 결과의 유효성을 판단하기 위해서 질의가 재평가되는 시간점이다. 두 개의 연속된 평가시간 점은 하나의 평가시간 인터벌을 구성하게 되고, 각각의 최근접 객체들은 탐색될 때마다 자신이 유효하게 존재하는 평가시간 인터벌과 함께 시간 리스트 TL에 {⟨a, [t1, t2]⟩, ⟨b, [t2, t3]⟩} 형태로 저장되었다가, 질의 마지막 시간에 결과로 반환된다.

CTNN 질의는 시작 시간에서 전체 데이터 객체와 질의점 사이의 거리 계산을 행한 후, 최소한의 추가 계산 작업만으로 다음 하위 시간 인터벌의 최근접 객체들을 선택한다. 단지 현재 p의 유효시간 인터벌 내에서 어떤 객체가 삽입되었거나 삭제되었을 때만 이 갱신으로 인해 유효시간 인터벌이 변경되는지 다른 최근접 객체가 선택되는 지를 체크하고 최소한의 거리 계산을 행하며, 나머지 평가시간 위에서는 질의와 교차하는 세그먼트 p', 또는 현재의 p와 교차하는 p'가 있을 때 p'로 최근접 객체 정보를 변경시켜주기만 하면 된다. CNN 질의는 어떤 객체의 속도나 방향이 변경될 때마다 거리 계산 함수를 재계산하고, 함수가 변경될 때마다 객체들과 질의 사이의 거리를 재계산하며 이들을 정렬한다. 또한 최근접 객체 선택과정에서 유효시간 인터벌을 결정하기 위한 분할점 계산 작업도 필요하다. CTNN은 이러한 불필요하고 반복적인 작업들을 줄이고, 간단한 계산을 통해 정확하고 연속적인 최근접 탐색을 가능하게 하였다.

4.2. Approximate CTNN 질의

실 세계 응용에서 어떤 사용자는 자신이 원하는 응답을 얻을 때, 결과의 정확도보다는 빠른 응답 시간을 더욱 중요하게 생각할 수도 있다. 즉, 약간의 오차가 허용되는 범위 내에서 빠르게 결과를 얻기를 원할 지도 모른다. 이런 경우 사용되는 것이 근사 질의 처리 기법이다. CTNN 질의는 선분 대칭 이동과 교차점, 변곡점에서의 결과 변경 여부 비교를 통해 사용자에게 정확한 결과를 제공하지만 세그먼트 수가 많아질 수록, 세그먼트 교차 수가 증가할 수록 대칭 이동 횟수나 평가시간 점 개수가 증가하기 때문에 빈번한 비교 작업을 행하게 된다. 이로 인해 응답 시간이 느려지는 것을 방지하기 위하여, 우리는 다음의 알고리즘 1와 같이, 좀 더 빠른 시간 안에 근사 정보를 얻을 수 있는 Approximate CTNN 기법을 제안한다.

알고리즘 1. Approximate_CTNN

Input: 데이터 객체 세그먼트 집합 seg, n개의 세그먼트(q1, ⋯, qn)로 구성된 질의 객체 궤적 q
Output: 시간 리스트 TL

//단계 수행 중 후보 객체가 여럿 존재하는 경우에는 항상 Calculate_Displacement( ) 적용
1. set_EvaluationTime() // 두 세그먼트의 실제 교차점만을 평가시간으로 선택
2. |qi − segi|tsi // qi의 시작 시간 tsi
2.1 mindist 가지는 최근접 객체 p 선택
2.2 TL = ⟨p , [ts , tei]⟩
3. 탐색원 h 그리기 (지름: |qi − p|tsi , 중심점: 선분qip의 중점)
4. h와 교차하는 ㅁp'가 존재하는지 검색
4.1 ㅁp'가 존재한다면,
4.1.1 교차시간 th 설정
4.1.1.1 h와 ㅁp'가 교차하는 두 점 중, 이후 시간을 가지는 교차점 선택

4.1.1.2 한 점만 교차한다면, 그 점을 교차점으로 선택

4.1.2 |qi − p'|th 〈 |qi − p|th 라면 TL = {〈p , [tsi , th]〉, 〈p' , [th , tei]〉}

4.1.3 아니면 TL = 〈p , [tsi , tei]〉

4.2 □p'가 존재하지 않는다면, TL = 〈p , [tsi , tei]〉

5. Until tei // □qi의 끝 시간

5.1 Find_NNObject()의 4단계 수행

5.2 □q에 의해 발생되지 않은 교차점 ti에 대해, 3~4단계 반복

6. Until te // q의 끝 시간, 즉 세그먼트 qn의 끝시간과 동일

6.1 각 질의 세그먼트 qi에 대해, tsi에서 3~4단계, tei까지 5단계 반복 수행

7. TL 반환

Approximate CTNN 질의는 선분 대칭이동 작업을 행하지 않으며, 두 세그먼트가 실제로 생성하는 교차점만을 평가시간으로 가진다. 따라서 평가시간 개수를 줄일 수 있으며, 이로 인해 최근접 객체 정보 변경 비교 횟수를 줄일 수 있다. 대신에 정확도를 보충하기 위하여 탐색원과 세그먼트의 교차점을 평가시간에 포함한다.

4.3. Exact CTNN 질의

Exact CTNN 기법의 기본 아이디어는 선분 세그먼트 대칭 이동과 탐색원생성 과정을 혼합한 것이다. 먼저 Basic CTNN 질의와 동일하게, 첫 번째 교차점이 존재하는 공간이 기준공간으로 선택되며, 질의 시작시간에서 Approximate CTNN 질의와 동일하게 모든 객체들과 질의점 사이의 거리를 계산한 후, 가장 작은 거리를 가지는 객체 p를 선택한다. 그리고 p가 이동공간 내의 객체라면 이를 기준공간으로 이동시키고, p와 기준공간 내 세그먼트간의 교차점을 탐색한다. p와 q 사이에 탐색원을 생성하고 이에 교차하는 다른 세그먼트가 존재하는지 체크한다. 만일 질의 한 세그먼트가 끝나기 전에 p가 다른 교차점과 만나거나 새로운 시작점 또는 끝점이 생긴다면 Find_NNObject()를 따라 처리한다. 그리고 p가 질의 세그먼트와 교차점을 생성하는 경우에만 교차점 이후 시간에 존재하는 p의 일부 세그먼트를 반대쪽 공간으로 대칭이동 시킨다. Exact CTNN 질의는 알고리즘 2과 같이 수행된다.

알고리즘 2. Exact_CTNN

Input: 데이터 객체 세그먼트 집합 seg , n개의 세그먼트(q1,…,qn)로 구성된 질의 객체 궤적 q

Output: 시간 리스트 TL

// 단계 수행 중 후보 객체가 여럿 존재하는 경우에는 항상 Calculate_Displacement( ) 적용

1. Set_EvaluationTime()

2. |qi − segi|tsi // qi의 시작 시간

2.1 mindist 가지는 최근접 객체 p 선택

2.2 TL = 〈p , [tsi , tei]〉

3. 탐색원 h 그리기 (지름: |qi − p|tsi , 중심점: 선분qip의 중점)

4. h와 교차하는 □p'가 존재하는지 검색

4.1 □p'가 존재한다면,

4.1.1 교차시간 th 설정

4.1.1.1 h와 □p'가 교차하는 두 점 중, 이후 시간을 가지는 교차점 선택

4.1.1.2 한 점만 교차한다면, 그 점을 교차점으로 선택

4.1.2 |qi − p'|th 〈 |qi − p|th 라면 TL = {〈p , [tsi , th]〉, 〈p' , [th , tei]〉}

4.1.3 아니면 TL = 〈p , [ts , tei]〉

4.2 □p'가 존재하지 않는다면, TL = 〈p , [ts , tei]〉

5. Until tei // □qi의 끝 시간

5.1 Find_NNObject()의 4.2 ~ 4.3단계 수행

5.2 ti시간에 IL내의 교차점 ip1가 존재할 때,

5.2.1 □q와 p'에 의해 발생된 교차점이라면,

5.2.1.1 ti시간 이후의 □p'를 반대쪽 공간(기준공간)으로 대칭이동

5.2.1.2 □p'와 기준공간 내 segi에 대해 Find_Intersection() 수행 ▸IL

5.2.2 □q에 의해 발생되지 않은 교차점이라면, ip1과 qi에 대해 3~4단계 반복

6. Until te // q의 끝 시간, 즉 세그먼트 qn의 끝시간과 동일

6.1 각 질의 세그먼트 qi에 대해, tsi에서 3~4단계, tei까지 5단계 반복 수행

7 TL 반환

V. 실험 및 성능 평가

이 절에서는 Approximate와 Exact CTNN 기법의 응답시간, 질의 결과 정확도 성능을 비교하였다. CTNN 질의는 연속 질의 개념과 궤적 질의 개념을 가지고 있고 연속 질의 개념 측면에서 비교될 수 있는 CNN 질의와 비교평가 하였다. 이동객체를 위한 CNN 질의는 k-최근접 질의를 위해 제안된 기법이지만, 이 실험에서는 CNN의 k 최근접 질의 처리 단계 중 가장 먼저 수행되는 분할 리스트를 결정하는 단계, 즉 최근접 객체가 갱신되는 시점을 결정하는 단계에 초점을 맞추어 비교평가 하였다.

5.1. 공간 분포 범위에 따른 응답시간

우리는 먼저 세그먼트의 공간 분포 범위와 Exact, Aproximate, CNN 기법들의 질의 응답시간에 따른 변화를 살펴보았다. 그림 3은 각 공간 분포에 따른 질의 응답시간을 나타내고 있다. 그림 3에서 알 수 있듯이 Exact, Approximate, CNN 기법은 모두 공간 범위에 따른 질의 응답 시간에 큰 영향을 받지 않는다. 특히, Exact, Aproximate 기법의 경우 그림 4에서 보여주듯이 추출한 교차점 개수가 공간 범위에 따라 크게 달라지지 않기 때문이다.

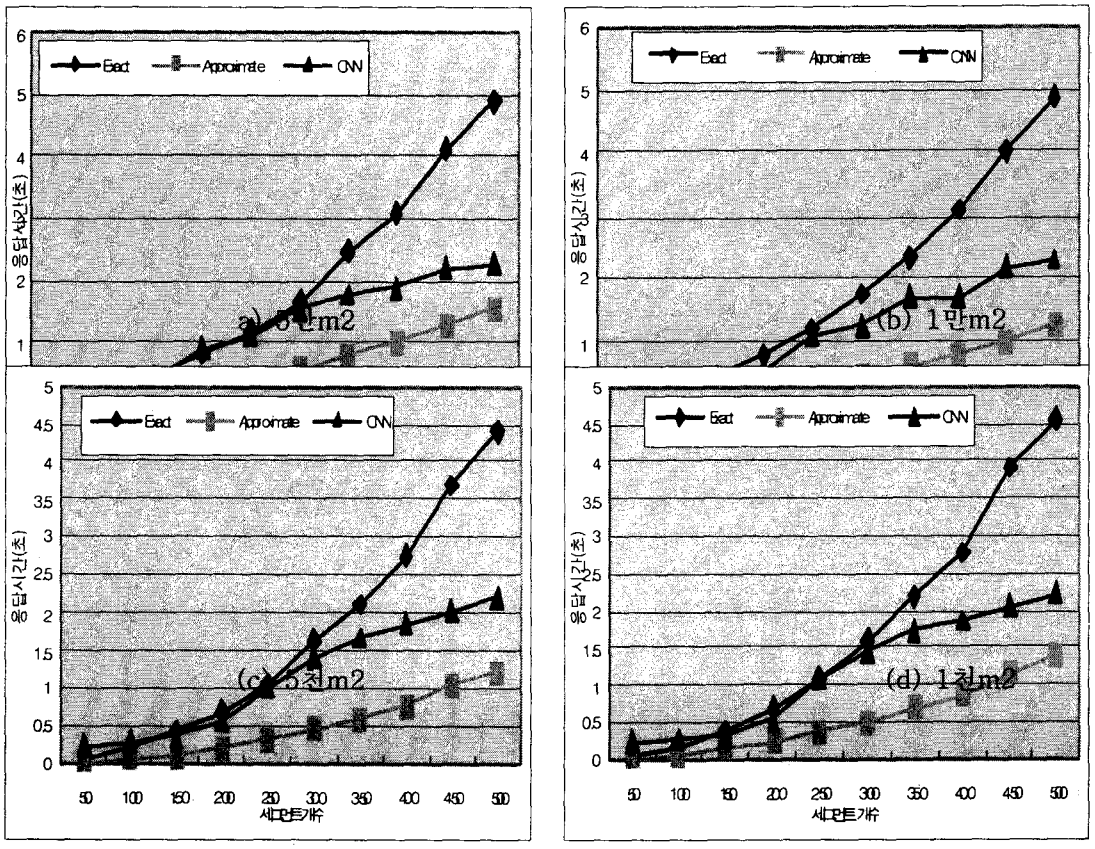


그림 3. 공간 범위 분포에 따른, 세그먼트 개수 vs. 평균 응답 시간

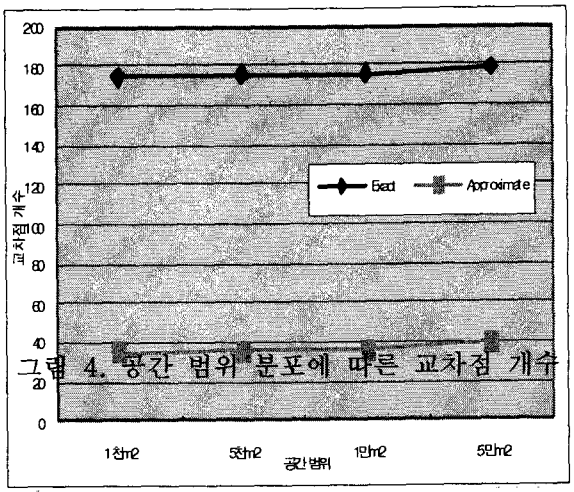


그림 4. 공간 범위 분포에 따른 교차점 개수

5.2. 정확도 비교

우리는 세그먼트의 수가 5, 10, 15, 20, 25, 30, 50, 100개 인 경우, 일정 공간 비율과 일정 시간 범위를 가지는 세그먼트들을 각각 임의로 생성한 후 Exact, Approximate, CNN 기법을 적용하였을 때 각 기법들이 어느 정도의 정확도를 보이는지 살펴보았다.

우리는 정확한 결과 값과 각 기법이 탐색해 낸 하위 시간 인터벌들 간의 개수를 비교하여 평가하였다. 그 이유는 인터벌 개수가 동일한 경우 시점 간의 차를 정확히 판별할 수 있기 때문이다. 따라서, 각 데이터 집합에 대해 단순 방법으로 최근접 객체를 탐색한 경우와 각 CTNN 기법이 탐색한 결과를 비교하고, 인터벌 개수가 실제 결과와 동일한 경우 시작 또는 끝 시간차가 1초 이내라면 정확히 일치하는 것으로 간주하였으며, 아니라면 인터벌 차이 수를 1로 설정하였다. 그리고 각 기법이 탐색해 낸 결과와 정확한 결과 사이의 인터벌 개수 차를 정확도 값으로 산출하였다.

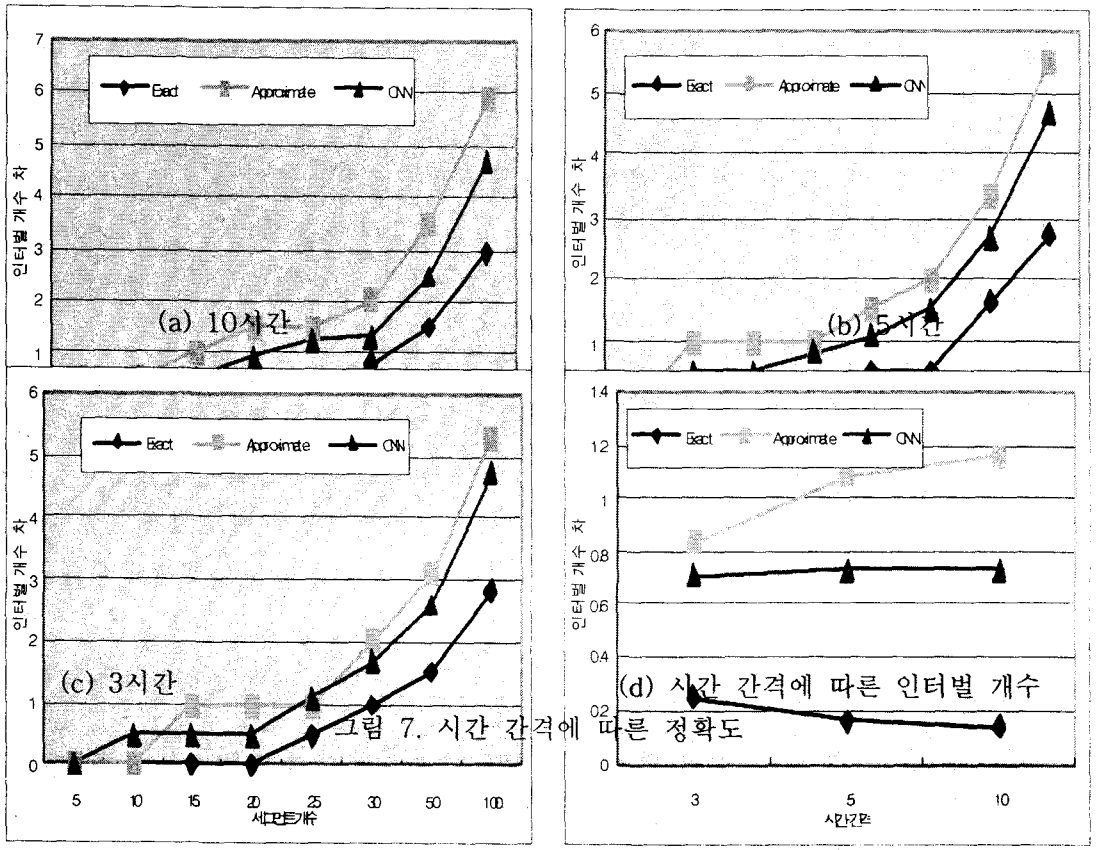


그림 7. 시간 간격에 따른 정확도

그림 7의 실험 결과에서 보여주듯이, 모든 시간 간격에 대해 대부분 같은 경향을 보였다. 즉, 각 기법들 모두 세그먼트의 개수가 증가할수록 인터벌 개수 차를 많이 나타내었다. Exact 기법의 경우 모든 경우 정확도가 높았으며, Approximate 와 CNN의 경우 CNN이 약간 좋은 성능을 보였다. 또한 시간 간격에 따른 전체적인 정확도를 비교해 보았을 때 그림 4(d)에서 보여주듯이 Exact 와 Approximate 기법의 경우 시간 간격에 따른 영향을 받는 반면, CNN은 시간 간격에 따른 영향을 받지 않는 것으로 나타났다. 이러한 이유는 세그먼트의 양 끝점 시간 간격에 따른 실험에서도 살펴보았듯이 각 세그먼트의 양 끝 시간이 정해진 경우, 일정 범위 내에서 세그먼트를 임의로 생성하면 각 세그먼트의 속도에 차이가 생기며, 따라서 세그먼트의 길이나 세그먼트들의 분포 밀도에 영향을 끼치기 때문이다.

Ⅵ. 결론

이 논문에서 우리는 질의와 데이터 객체가 모두 이동객체인 경우에 가장 유용하게 사용될 수 있는 새로운 최근접 질의 처리 기법들을 제안하였다. 기존의 연구들은 데이터와 질의의 궤적 정보를 전혀 고려하지 상태에서 최근접 객체를 선택하기 때문에 부정확한 결과를 보일 수 있다. 또한 선택된 최근접 객체 정보가 빠른 시간 안에 변경될 수도 있기 때문에 잦은 갱신을 발생시킬 수 있다. CTNN 기법은 이런 문제점들을 해결하기 위하여 궤적 최근접 질의 개념을 사용하였으며, 변위 계산을 통해 질의 궤적과 가장 가까운 위치를 유지하면서 움직이는 객체를 최근접 객체로 선택하도록 하였다. 또한 최근접 객체가 변경되는 연속적인 시간을 탐색해내기 위하여 객체 세그먼트들의 변곡점과 교차점 위에서 결과 변경 여부를 비교하였다. 우리는 이와 같은 궤적 최근접 질의를 수행하기 위한 기본적인CTNN 알고리즘과 함께, 응답 시간을 줄이기 위한 Approximate CTNN 기법과 정확도 향상을 위한 Exact CTNN 기법의 알고리즘을 제안하였다. 그리고 각 기법들이 세그먼트의 공간 범위나 시간 범위에 따라 얼마만큼의 응답 시간 차이를 보이는지, 정확도 차이를 보이는지 실험을 통해 평가하였다. 실험 결과에서는 Exact CTNN 기법의 경우 높은 정확도를 갖는 반면, 응답시간에 있어서는 조금 낮은 성능을 보였다. 또한 Approximate CTNN 기법의 경우 빠른 응답시간을 보인반면, 정확도 측면에서는 Exact CTNN 보다 낮은 성능을 보였다. 따라서, 우리는 실험을 통해 정확한 답을 원하는 경우에는 Exact CTNN 기법을 사용하고, 빠른 응답을 원할 경우는 Approximate CTNN 기법을 사용할 수 있음을 보였다.

향후연구에서는 인덱스 구조를 이용한 효율적인 CTNN 기법 처리에 관한 연구를 수행할 것이며, 또한 이동 영역 객체에 대한 CTNN 기법에 관한 연구도 수행할 것이다.

참고문헌

1. A. Prasad Sistla, Ouri Wolfson, Sam Chamberlain, Son Dao, "Modeling and Querying Moving Objects", ICDE 1997, pp.422~432
2. Flip Korn, Nikolas Sidiropoulos, Christos Faloutsos, Eliot Siegel, Zenon Protopapas, "Fast Nearest Neighbor Search in Medical Image Databases", VLDB 1996, pp.215~226
3. Like Gao, Zhengrong Yao, Xiaoyang Sean Wang, "Evaluating Continuous Nearest Neighbor Queries for Streaming Time Series via Pre-Fetching", CIKM 2002, pp.485~492
4. Nick Roussopoulos, Stephen Kelley, Fredeic Vincent, "Nearest Neighbor Queries", SIGMOD Conference 1995, pp.71~79
5. Apostolos Papadopoulos, Yannis Manolopoulos, "Performance of Nearest Neighbor Queries in R-tree", ICDT 1997, pp.394~408
6. Stefan Berchtold, Bernhard Ertl, Daniel A. Keim, Hans-Peter Kriegel, Thomas Seidl, "Fast Nearest Neighbor Search in High-Dimensional Spac", ICDE 1998, pp.209~218
7. Cui Yu, Beng Chin Ooi, Kian-Lee Tan, H.V. Jagadish, "Indexing the Distance : An Efficient Method to KNN Processing", VLDB 2001, pp.421~430
8. Danxhou Liu, Ee-Peng Lim, We Keong Ng, "Efficient k Nearest Neighbor Queries on Remote Spatial Database Using Range Estimation", SSDBM 2002, pp.121~130
9. David A. White, Ramesh Jain, "Similarity Indexing with the SS-tree", ICDE 1996, pp.516~523
10. Norio Katayama, Shin'ichi Satoh, "The SR-tree : An Index Structure for High-Dimensional Nearest Neighbor Queries", GISMOD Conference 1997, pp.369~380
11. Suni Araym, David M. Mount, Nathan S. Netanyahu, Ruth Silverman, Angela Y. Wu, "An Optimal Algorithm for Approximate Nearest Neighbor Searching Fixed Dimensions", JACM 1998, pp.891~923
12. Piotr Indyk, Rajeev Motwani, "Approximate Nearest Neighbors : Towards Removing the Curse of Dimensionality" , STOC 1998, pp.604~613
13. King-Ip, Congjun Yang, "The ANN-tree : An Index for Efficient Approximate Nearest Neighbor Search", DASFAA 2001, pp.174~181
14. Stephan Volmer, "Fast Approximate Nearest Neighbor Queries in Metric Feature Spaces by Buoy-Indexing", VISUAL 2002, pp.36~49
15. George Kollios, Dimitrios Gunopulos, Vassilis J. Tsotras, "Nearest Neighbor

Queries in a Mobile Environment", Spatio-Tempolra Database Management 1999, pp.119~134

16. Rimantas Benetis, Christian S. Jensen, Gytis Karciauskas, Simonas Saltenis, "Nearest Neighbor and Reverse Nearest Neighbor Queries for Moving Objects", IDEAS 2002, pp.44~53
17. Simonas Saltenis, Christian S. Jensen, "Indexing the Position of Continuously Moving Object", SIGMOD Conference 2000, pp.331~342
18. Yufei Tao, Dimitris Papadias, "Spatial Queries in Dynamic Environments", TODS 2003
19. Jose Moreira, Cristina Ribeiro, Jean-MarcSaglio, "Representation and Manipulation of Moving Points: An Extended Data Model for Location Estimation" , Cartography and Geographic Information Systems(CaGIS), ACSM, Vol.26, No.2, April 1999
20. Kriengkrai Porkaew, Iosif Lazaridis, Sharad Mehrotra, "Querying Mobile Objects in Spatio-Temporal Databases", SSTD 2001, pp.59~78
21. Yannis Theodoridis, Mario A. Nascimento, " Generating Spatiotemporal Datasets on the WWW", SIGMOD 2000, pp.39~43
22. Michalis Vazirgiannis, Yannis Theodoridis, Timos K. Sellis, "Spatio-Temporal Composition and Indexing for Large Multimedia Applications", Multimedia Systems 1998, pp.284~298
23. Dieter Pfoser, Yannis Theodoridis, Christian S. Jensen, "Indexing Trajectories of Moving Point Objects", CHOROCHRONOS Technical Report CH-99-03, October, 1999
24. Dieter Pfoser, Christian C. Jensen, Yannis Theodoridis, " Novel Approaches in Query Processing for Moving Objects", CHOROCHRONOS Technical Report CH-00-3, February, 2000
25. Dieter Pfoser, Yannis Theodoridis, " Generating Semantics-Based Trajectories of Moving Objects", International Journal of Computers, Environment and Urban Systems, special issue, accepted. Elsevier. May, 2000
26. M. de Berg, M. van Kreveld, M. Overmars and O. Schwarzkopf, Computational Geometry Algorithms and Applications. Springer-Verlag, March 1997.