

이동 객체 데이터베이스를 위한 거리기반 인덱싱 전략*

김승현^o 송문배^{*} 남성현^{*} 황종선^{*} 손진곤^{**}
^{*}고려대학교 컴퓨터학과, ^{**}한국방송통신대학 컴퓨터학과

{shkim, mbsong, shnam, hwang}@disys.korea.ac.kr, {jgshon@mail.knou.ac.kr

Distance-based Indexing Strategy for Moving Objects Database

Seung-Hyun Kim^o Moon-Bae Song^{*} Sung-Hun Nam^{*} Chong-Sun Hwang^{*} Jin-Gon Shon^{**}

^{*}Department of Computer Science and Engineering, Korea University.

^{**}Department of Computer Science, Korea National Open University.

요 약

이동 기술의 발전으로 위치와 이동 패턴이 연속적으로 변화하는 객체가 증가하였다. 시간에 따라 객체가 이동하면서 그 위치와 모양이 연속적으로 변화하는 것을 이동 객체라 한다. 과거의 DBMS는 연속적으로 이동 정보가 변화하는 객체를 관리하기 위해 빈번한 DB 갱신이 이루어져야 하는 문제점이 있다. 객체의 이동 정보는 연속적으로 변화한다. 이러한 정보를 저장하는 매체인 이동 객체 데이터베이스의 갱신 횟수는 매우 중요하다. 본 논문에서 객체는 일반적으로 일정한 패턴으로 이동한다고 가정한다. 이러한 객체의 일정한 이동 패턴을 이용하여, 이동 객체 데이터베이스의 갱신 횟수를 최소화하면서, 효과적으로 인덱싱 하는 전략을 제시한다. 또한 질의에 대한 객체 위치의 오차를 최소화하여 빈번해 주는 질의 처리 기법을 제안한다.

1. 서 론

이동 기술의 발전으로 연속적으로 위치와 이동 패턴이 변화하는 이동성을 가진 객체가 등장했다. 시간에 따라 객체가 이동하면서 그 위치와 객체의 모양이 연속적으로 변화하는 것을 이동 객체(Moving Object, MO)라고 한다[5]. 이러한 이동 객체의 위치 정보를 정확하게 제공하는 기술은 중요하다. 이러한 기술은 비행기, 선박, 차량의 항법장치에 GIS와 함께 GPS 응용 분야에 적용되고 있다. 과거의 DBMS는 연속적으로 변화하는 수많은 이동 객체의 위치 정보를 관리하기 위해서는 빈번한 DB 갱신이 발생하는 문제점이 있다. 또한, 과거 및 미래에 대한 이동 객체의 위치 정보를 관리 할 수 없는 문제점이 있다. 이러한 문제점으로 인해 이동 객체를 관리하기 위한 새로운 데이터베이스 모델의 필요성이 증가하고 있다[1]. 이동 객체 데이터베이스(Moving Objects Database; MODB)는 대상 객체들의 위치 정보를 관리하며 사용자 질의의 처리하는 역할을 수행한다. 이동 객체가 MODB에 정보를 제공하는 수단은 무선 링크 환경에서 이루어진다. 본 논문에서는 MODB의 갱신 횟수를 최소화하면서, 효과적으로 인덱싱 하는 방법을 제안한다. 또한 질의에 대한 이동 객체의 위치의 오차를 최소화하여 빈번해 주기 위한 효율적인 질의 처리 기법을 제안한다. 본 논문 구성은 다음과 같다. 2절에서는 관련 연구에 대하여 설명한다. 3절에서는 이러한 문제점을 해결하기 위한 제안 기법과 시스템 모델을 소개한다. 4절에서는 제안 기법의 인덱싱과 질의 처리 절차를 설명한다. 그리고 5절에서는 시뮬레이션을 통하여 성능 분석결과를 보여준다. 마지막으로 6절에서 결론 및 향후 연구 과제를 언급하고자 한다.

2. 관련 연구

과거 DBMS의 문제점은 이동 객체 데이터 모델을 제안한 연구에 의해서 어느 정도 해결되었다. DOMINO (Database for Moving Object) 프로젝트의 MOST (Moving Objects Spatial Temporal)는 선형 함수를 이용하여 이동 객체의 궤적을 모델링 하였다[1]. 이를 응용한 연구로 TPR-tree, R*-tree 이용한 인덱싱 기법이 연구되었다. MOST의 경우 이동 객체가 직선으로 움직일 경우에는 효과적인 방법이다. 그러나 그렇지 않은 이동 패턴일 경우, MODB 갱신 연산 횟수가 증가하는 문제점이 있다.

이러한 현상에 대하여 공간을 일정한 영역으로 분할하여 관리하는 해싱 기법이 연구되어 어느 정도 해결하였다[4]. 그러나 해싱 기법은 할당된 영역의 경계 지점을 이동하는 이동 객체의 경우 빈번한 MODB 갱신을 유발하는 단점이 있다. 또한 동일한 이동 패턴인 경우라도, 이동 객체가 할당된 영역 중 어느 위치에서 이동하는가에 따라 갱신 횟수가 다르게 나타나는 현상을 보인다.

3. 제안 기법과 시스템 모델

3.1. 제안 기법의 가정

실세계의 다양한 이동 패턴은 실측 데이터를 구하기 어렵기 때문에 명확하게 분석하기 힘들다. 하지만 본 논문에서는 기존 논문에서 언급한 내용을 기반으로 다음과 같이 실세계 객체의 일반적인 특성을 가정하고자 한다. 첫째, 일반적으로 객체는 목적지를 향하여 이동하는 특성을 가진다[2]. 둘째, 객체는 이러한 목적지로의 경로 중 최적의 경

로를 선택하는 경향이 있다[2]. 마지막으로 이동 객체는 동일한 이동 패턴을 반복할 확률이 높다[7].

객체는 일반적으로 자신의 평균 속력을 기반으로 일정 거리를 이동한다. 속력이 빠른 객체는 속력이 느린 객체에 비해 단위 시간에 많은 거리를 이동하게 된다. 따라서 객체의 속력에 비례하는 이동 거리마다 위치 정보를 갱신한다고 가정하면, 속도가 빠른 객체는 다소 느린 객체보다 더 많은 거리를 움직인 후 자신의 위치를 갱신하게 된다. 반면, 느린 객체는 속력이 빠른 객체보다 짧은 거리를 이동한 후 갱신하게 된다. 그러나 이러한 위치 정보 갱신 접근은 모든 객체의 위치 정보의 갱신 주기를 동일하게 해준다.

이러한 이동 객체는 크게 점 객체와 영역 객체로 나눌 수 있다[5]. 점 객체는 객체 자신의 크기는 변화하지 않거나 크기 변화가 무의미한 객체를 말하며, 자동차를 예로 들 수 있다. 반면 영역 객체는 자신의 크기와 이동 패턴이 모두 변화하는 객체를 말하고, 태풍을 예로 들 수 있다. 이러한 영역 객체는 주로 영역의 변화가 중요한 질의 대상이다. 그러나 실세계의 이동 정보를 필요로 하는 객체의 대부분은 객체의 크기 변화에는 관심이 없다. 즉 객체의 위치에 기반한 서비스가 추가된다. 특히 이동 객체의 위치를 서비스 해주는 시스템은 객체의 위치만 정확히 서비스 해주면 된다. 그러므로 본 논문에서의 이동 객체는 점 객체로 한정한다.

또한 객체는 연속적으로 변화하는 자신의 위치와 이동 패턴을 GPS나 무선망을 통하여 객체 자신이 획득할 수 있다. 즉, 객체는 자신의 이동 정보(속도, 방향 등)를 스스로 연산해 낼 수 있으며, 그 정보를 필요로 하는 시스템에 제공할 수 있다.

마지막으로 실세계 공간 표현은 3차원으로 표현한다. 그러나 3차원 고려사항 중 수치 성분은 특수한 응용분야(군사분야 등)에서만 필요한 속성이다. 그러므로 본 논문에서는 2차원 평면상에서 이동하는 객체만을 가정한다.

3.2. 거리 기반 인덱싱 기법

객체는 단위 시간 안에 일정한 거리를 이동한다.

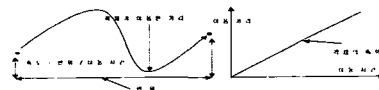


그림 1. 객체의 속도와 속도

그림 1에서 객체가 이동한 총 거리를 $Distance_{total}$, 객체가 이동하면서 경과된 시간을 $Time_{total}$ 이라 하면, 평균속력 $Speed_{average}$ 는 아래 식 1을 만족한다.

$$Speed_{average} = \frac{Distance_{total}}{Time_{total}} \quad (1)$$

객체는 이러한 평균 속력을 반영하는 거리를 이동한다. 즉 단위 시간 당 객체가 움직인 거리는 아래 식 2를 만족한다.

$$Distance_{단위시간} = speed_{average} \times 단위시간 \quad (2)$$

$Distance_{단위시간}$ 는 단위 시간당 객체가 평균적으로 움직인 거리와 동일하다. 즉 이 거리는 평균 이동 거리 $Distance_{average}$ 로 표현한다고 가정하면, $Distance_{average}$ 는 식 3에서의 같이 $Speed_{average}$ 를 반영하는 통계적 상수 값이다.

* 이 논문은 2001년도 한국학술진흥재단의 지원에 의하여 연구되었음. (KRF-2001-041-E00239)

$$\sum_{i=1}^{Time_{total}} Distance_{average_i} = Distance_{total} \quad (3)$$

평균 이동 거리는 평균 속력을 반영하는 통계적 상수 값이다. 시스템 초기에 평균 이동 거리 속성을 상수 값으로 설정한다. 거리 기반의 접근 방식은 인덱싱 주기를 일정하게 해준다. 속력이 빠른 객체와 상대적으로 속력이 느린 두 객체는 아래 식 4를 만족하는 동일한 주기로 인덱싱이 수행된다. 제안 기법에서 평균 이동 거리 $Distance_{average}$ 속성을 인덱싱하는 시점의 기준으로 적용한다. 즉 인덱싱하는 시점에 MODB에 객체의 이동 정보를 제공하는 방식이다.

$$\frac{Distance_{average\ fast}}{Speed_{average\ fast}} = \frac{Distance_{average\ slow}}{Speed_{average\ slow}} \quad (4)$$

제안 기법에서 초기에 설정된 평균 이동 거리를 반영하는 상수 값을 AD(Averaged Distance that object moves)이라 가정한다. AD는 객체의 위치 정보를 MODB에 저장할 시점을 결정한다. 객체의 위치를 인덱싱 하는 방법은 MODB에 마지막으로 저장된 위치에서 객체가 이동한 변위 (displacement)가 AD를 벗어나는 위치인지 아닌지를 비교하여 수행한다. 변위는 속도를 고려한 값으로, AD와 비교하는 길이는 이동 거리가 아니라 변위이다. AD 상수 값은 평균 이동 거리의 배수 값으로 설정한다. 평균 이동 거리의 1배수인 AD 상수 값은 평균 속력을 반영하는 평균 이동 거리와 일치한다. AD는 시뮬레이션을 통해 최적의 배수 값을 구하여 AD 상수 값으로 설정한다.

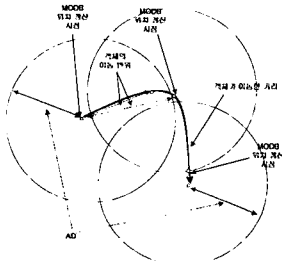


그림 2. 거리 기반 인덱싱 기법

객체가 이동한 변위를 $\vec{\Delta D}$ 으로 표현하면, $\vec{\Delta D}$ 는 객체의 현재위치 - 최근 MODB 갱신 위치이다. 인덱싱 알고리즘은 다음과 같이 수행된다. 만약 $AD < \vec{\Delta D}$ 를 만족하면 인덱싱이 수행된 정보를 저장한다. 즉 MODB에 저장되어 있는 해당 MO의 위치 정보가 갱신된다. 반면, $AD > \vec{\Delta D}$ 인 경우는 MODB의 갱신이 수행되지 않는다. $\vec{\Delta D}$ 의 연산은 이동 객체가 수행한다.

AD는 이전 갱신 위치가 (X, Y) 일 경우, 이동 객체의 위치를 중심으로 한 원의 형태로 표현 될 수 있다. 이동 객체의 위치를 $MO(X_{mo}, Y_{mo})$ 라 했을 경우 AD는 아래 식 5를 만족한다.

$$(X - X_{mo})^2 + (Y - Y_{mo})^2 = AD^2 \quad (5)$$

거리 기반의 인덱싱 기법은 MODB의 갱신 횟수를 줄이면서, 효과적으로 객체 위치 정보를 인덱싱 하고자 하는데 근본적인 목적이 있다. 인덱싱 방법은 객체의 이동 경로 중 MODB에 객체의 위치 정보를 갱신하는 시점을 중심으로 AD를 반지름으로 하는 원을 그려나가며 그림 2와 같이 인덱싱 과정을 수행한다.

3.3. 직선 이동 패턴을 고려한 인덱싱 기법

3.2절에서 제시한 거리 기반 인덱싱 기법은 객체가 직선 이동할 경우 AD를 이용하여 오차가 적은 위치 정보를 연산해 낼 수 있는데도 불구하고, AD 범위를 벗어나는 때마다 불필요한 MODB 갱신을 수행한다. 이에 객체의 직선 이동 패턴을 고려한 인덱싱 기법을 제안한다.

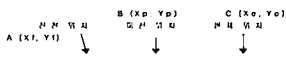


그림 3. 직선 이동 패턴

객체가 직선으로 이동하는 경우는 위 그림 3과 같이 이동 궤적중 세 점이 일직선상에 놓이는 경우이다. 전전 위치를 $A(X, Y)$ 전 위치를 $B(X, Y)$, 현재 위치를 $C(X, Y)$ 라 하면, 아래 식 6을 만족하는 경우 객체는 직선 이동 패턴으로 간주한다.

$$\frac{(C - A)^2 + (C - B)^2}{(C - A)^2} = 1 \quad (6)$$

위 식 6을 만족하는 경우 MODB에 위치 갱신 정보를 제공하지 않는다. AD에 의해 객체의 최소의 오차를 가진 위치를 연산해 낼 수 있기 때문이다. 객체의 직선 이동 여부를 판단하는 연산은 객체 스스로 한다.

객체가 직선 이동할 경우 아래 식에 의해 위치를 연산한다. AD를 벗어난 위치에 객체가 위치한 경우, 객체가 자신이 직선 패턴으로 이동하는지, 아닌지에 대한 여부를 MODB에 알려 준다.

$$\text{연산된위치} = \text{MODB에 최근 갱신된 위치} + (\text{Distance}_{average} \times (\text{현재 시간} - \text{최근 MODB 갱신 시간})) \quad (7)$$

3.4. 시스템 모델

제안 기법의 시스템 모델은 다음과 같이 분류할 수 있다. 객체의 이동 정보를 실시간으로 처리하는 지역 처리기(Region Processor; RP)와 AD를 고려하여 제공된 객체의 위치 정보를 저장하는 이동 객체 데이터베이스(MODB)가 있다. 또한, 질의 결과로 반환되기 전 객체 위치 정보의 오차를 줄이기 위한 선-질의 처리기(Pre-Query Processor; Pre-QP)가 있다. 오차가 없는 질의 결과(Query Result without error; QR w/o)는 객체의 위치 정보 전달 수단인 무선 네트워크 상의 지연과, 샘플링 단위 시간의 지연을 배제한 결과로 가정한다. 이 위치를 현재 위치라 가정한다. MODB에 저장된 실제 위치와의 오차와 지연요소를 모두 포함한 질의 결과물 오차를 가진 질의 결과(Query Result with error; QR w)라고 가정한다. 본 시스템 모델은 QR w와 QR w/o를 모두 질의 결과로 반환할 수 있다. 질의 시 질의 처리비용을 질의자가 상대적으로 많이 부담할 경우는 QR w/o를 질의 결과로 반환하고, 그렇지 않은 경우 QR w를 반환한다.

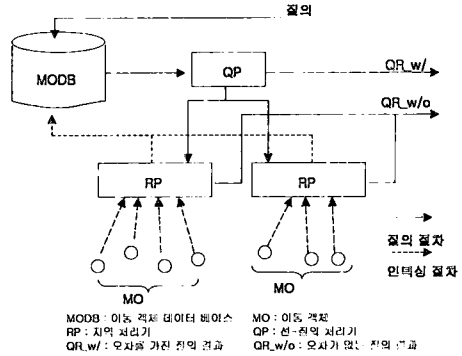


그림 4. 시스템 모델

3.4.1 지역 처리기 (RP)

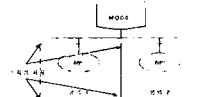


그림 5. 지역 처리기

MOST를 포함한 기존의 연구에서는 모든 객체의 이동 정보를 하나의 MODB에 저장하는 시스템 모델을 제시했다. 이러한 중앙 집중식 시스템은 MODB의 부하로, 효율적이지 못하다. 또한, MODB에 결합이 발생할 때 시스템 전체의 가용성이 떨어지는 단점이 있다. 이에 해상 기법에서는 MODB에 영역 아이디만 저장하고, 객체의 위치 정보는 선-처리기에 저장하는 방식을 제안하였다[4]. 그러나 해상 기법에서 제안한 모델은 단순히 이동 객체의 수를 절반으로 나누어 관리하는 시스템으로 객체의 이동성을 전혀 반영하지 못한다. 또한, MODB에 영역 정보만 저장되므로, MODB의 정보만으로는 객체의 위치를 전혀 예측할 수 없다. 이에 제안한 시스템 모델에서 RP는 자신이 관할하는 영역에 한 개씩 위치한다. 넓은 영역에 존재하는 많은 이동 객체를 임의의 크기의 영역으로 분할하여 관리한다. 즉 각 영역에 존재하는 객체를 그 영역을 담당하는 RP가 나누어 관리한다. 그리고 각각의 지역을 관할하는 동일한 기능의 RP가 AD를 고려하여 MODB에게 분산 관리된 객체의 위치 정보를 제공하게 된다. 많은 이동 객체는 RP가 관할하는 영역에서 분산 관리된다. 이동 객체는 자신의 위치 정보를 관할하는 RP에게만 제공하게 된다. 이러한 접근 방식은 MODB가 모든 객체의 정확한 위치 정보를 관리해야 하는 부하를 분산시켜줌으로써, 효율적이며 확장이 용이한 위치 관리 시스템으로서의 장점을 갖는다. MODB에는 오차를 가진 객체의 위치 정보가 저장되고, RP에는 최소한의 오차를 지닌 현재 위치가 저장되는 시스템 모델이다.

RP에는 영역의 물리적 위치 정보인 영역의 꼭지점 좌표 값 $\langle P_{x1}, x1, P_{y1}, y1 \rangle$ 속성과 AD의 상수 값 $\langle AD \rangle$ 속성을 가지고 있다. 또한, 현재 객체의 위치 $\langle P_{x, current}, y \rangle$ 와 객체의 아이디 $\langle ID_{mo} \rangle$ 속성을 가지고 있다. RP는 MODB의 갱신 횟수를 줄이기 위해 AD를 벗어나는 객체의 정보만을 MODB에 제공하는 역할을 한다. [MO 아이디], [현재 위치 (Xc, Yc)], [AD], [관할 영역 정보]

그림 6. RP 속성

3.4.2 이동 객체 데이터베이스 (MODB)

MODB에는 갱신된 객체의 위치 < $P_{modb-lastest}$ > 와 AD 상수 값이 저장된다. < 갱신 시간 > 속성은 객체의 위치 정보가 갱신된 시간이다. < 직선 이동 여부 > 속성은 객체가 직선 이동 패턴일 경우 < 0 > 으로 설정되고, 아닐 경우 < 1 > 으로 설정된다.

MO 아이디	갱신 위치 (Xu,Yu)	AD	직선이동여부	갱신 시간
--------	---------------	----	--------	-------

그림 7. MODB 속성

3.4.3 선 질의 처리기 (Pre-QP)

Pre-QP는 질의 결과 처리시 MODB에 저장되어 있는 위치와 실제 위치와의 오차를 최소화 해준다. 또한 질의에 해당하는 객체의 네트워크 지연과 단위 샘플링 시간을 배제한 정확한 위치 값을 반환하기 위해, RP를 선별해 주는 기능도 한다. 해싱 기법에서는 별도의 아이디를 각 영역을 관리한다[4]. 그러나 제안 기법에서는 별도의 아이디를 관리 필요 없다. Pre-QP에 의해 연산된 위치가 소속된 좌표 값을 영역 정보로 가진 해당 객체를 관리하는 RP를 찾아 낼 수 있다. RP를 찾아내어 저장되어 있는 객체의 현재 위치를 반환한다.

4. 인덱싱과 질의 절차

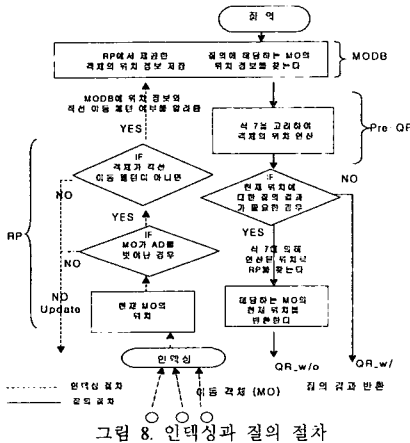


그림 8. 인덱싱과 질의 절차

위 그림 8의 점선 부분은 인덱싱 과정을 보여주고 있고, 실선 부분은 질의 처리 과정을 보여 준다. 인덱싱 과정에서 객체의 현재 위치 $P_{rp-current}$ 가 AD를 벗어난 경우, 객체의 직선 이동 패턴 여부만 판단한다. 직선 이동이 아닐 경우 MODB에 위치 갱신 정보를 보낸다. 즉 해당 객체의 MODB에 저장되어 있는 위치 정보가 갱신된다. 질의 처리시 식 7에 의해 저장되지 않은 객체의 위치 정보를 연산하여 질의 결과로 반환한다. 질의 결과는 QR w/o와 QR w/로 반환할 수 있다. 질의 처리비용은 질의 한 측에서 부담한다. 상대적으로 질의 처리비용을 많이 부담하는 측에는 QR w/o을 질의 결과로 반환받게 된다. 이러한 제안된 인덱싱 기법과 질의 처리 절차는 MODB의 갱신 횟수를 최소화하는 인덱싱을 수행한다. 또한, 효과적이며 질의 처리 과정을 통하여 이동 객체에 대한 위치의 오차를 최소화하여 질의 결과도 반환해 준다.

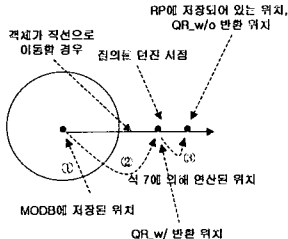


그림 9. 질의 결과 반환

5. 성능 분석 및 결과

일반적으로 시공간 시뮬레이션은 실제 데이터나 확률적인 데이터를 이용한다. 본 논문의 성능분석에서는 이동 객체의 초기 위치 및 이동 좌표는 확률적인 데이터인 Gaussian, Zipf, Uniform 분포를 이용하여 생성하였다[4][8]. 각각의 분포에 대한 속성값은 다음과 같이 초기화 하였다. 초기 생성 위치 최대/최소 범위 = 0.0-1.0, Mean = 0.5, Sigma = 0.1, P = 0.1, 다음 이동 위치 생성시 최대/최소 범위 =

-0.01-0.01, Mean = 0.0, Sigma = 3.01, P = 0.01[4][8]. 성능분석 결과 그래프는 1000초 동안 10,000개의 객체를 동시에 움직여 발생하는 MODB의 갱신 횟수와 실제 위치와 MODB에 저장되어 있는 위치 오차값을 산출하여 보여준다.

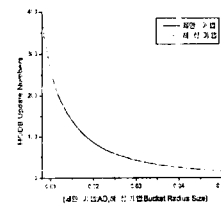


그림 10. MODB 갱신 횟수

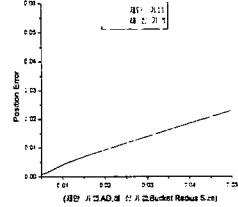


그림 11. 위치 오차

그림 10 그래프의 X좌표는 제안 기법의 AD값과 해싱 기법의 한 바킷의 한 번의 1/2값을 나타낸다. Y좌표는 MODB의 갱신 횟수를 보여준다. 그림 10 그래프에서 해싱 기법 보다 제안 기법의 경우 X좌표 값이 증가할수록 더 낮은 MODB 갱신 횟수를 나타낸다. 또한 다음 위치 생성 최대 범위보다 작은 X값의 경우 큰 X값의 경우보다 상대적으로 MODB 갱신 횟수가 빈번하게 일어난다는 것을 알 수 있다. 그러므로, 제안 기법의 AD 값은 다음 이동 위치 생성 최대범위 이후의 X값으로 설정해야 한다.

그림 11 그래프의 X좌표는 그림 10과 동일한 값을 나타내며, Y좌표는 MODB에 저장되어 있는 위치와 이동 객체의 실제 위치와의 거리의 오차를 나타낸다. 그림 11 그래프에서 X좌표값이 증가함에 따라, 제안 기법보다 해싱 기법의 경우 실제 위치와의 오차가 더 많이 나는 것을 알 수 있다. 성능 분석 결과 그래프의 X값의 증가 현상으로 인한 MODB의 갱신 횟수의 감소는 위치 오차의 증가 현상을 가져오는 것을 알 수 있다. 그러나 제안 기법에서 해싱 기법보다 갱신 횟수의 감소에 비례하는 오차 증가율이 해싱 기법보다 적은 것으로 나타났다.

6. 결론 및 향후 연구 과제

본 논문에서 제안한 AD를 이용한 제안기법의 장점은 다음과 같다. 첫째, MODB의 갱신 횟수를 줄이면서, 효과적으로 객체의 위치를 인덱싱하고 질의에 대하여 처리할 수 있게 해준다. 둘째, 해싱 기법에서 동일한 이동 패턴임에도 불구하고, 그 이동 패턴이 영역 중 어느 위치에 위치하여 있는가에 따라 인덱싱 성능 결과가 다르게 나오는 문제점을 해결하였다. 제안기법은 이동 패턴이 동일하면 항상 일정한 성능 결과를 보여 준다. 셋째, 객체의 현재 위치를 고려하지 않고, 영역 아이디를 이용하여 위치 정보를 관리하는 방식은 MODB에 저장된 정보만으로 객체의 위치를 전혀 예측할 수 없는 문제점을 해결하였다. 넷째, MOST의 경우에는 직선 이동 패턴이 긴 시간 지속 될 경우 유리한 기법이지만, ZigZag 이동 패턴과 같이 방향이 빈번하게 바뀌는 이동 객체의 경우, DB 갱신이 많이 일어나는 문제점이 있다. 본 제안 기법에서는 직선 이동과 방향이 빈번하게 변화하는 이동 패턴 모두에 대해 고려가 이루어 졌다.

향후 연구과제로는 이동 객체의 위치 관리 기법에 계층적 기법을 적용하는 연구가 진행 중이다. 또한 최근 객체의 이동 패턴을 예측하는 분야가 중요시되고 있다. 이에 MODB의 < 시간 > 속성을 이용하여 이동 패턴을 예측하는 연구가 진행 중이다. 또한, RP가 관리하는 영역 사이를 이동하는 핸드오프상황에 대한 연구가 진행 중이다.

7. 참고문헌

- [1] Oun Wolfson, Bo Xu, "Moving Objects Databases : Issue and Solutions," International Conference on Scientific and Statistical Database Management, pp. 111-122, 1998.
- [2] Jean Marc saglio, Jose Moreira, "Oporto:A Realistic Scenario Generator for Moving Objects," Geoinformatica, pp. 71-93, 2001.
- [3] Simonas Salteris, Christian S. Jensen, "Indexing the positions of Continuously Moving Objects," ACM SIGMOD International Conference on Management of Data, pp. 331-342, 2000.
- [4] Zhexuan Song, Nick Roussopoulos, "Hashing Moving Objects," International Conference on Mobile Data Management, pp. 161-172, 2001.
- [5] Ralf Hartmut Güting, "A Foundation for Representing and Querying Moving Objects," ACM Transactions on Database Systems, pp. 1-42, 2000.
- [6] Georce Kollios, Dimitrios Gunopulos, Vassilis J. Tsotras, "On Indexing Mobile Objects," ACM Symposium on principles of Database systems, pp. 261-272, 1999.
- [7] Ming Hour Yang, Lien Wu Chen, Yu Chee Tseng, "A Traveling Salesman Mobility Model and Its Location Tracking in PCS Networks," International Conference on Distributed Computing Systems, pp. 517-523, 2001.
- [8] Dieter Pfoser, Yannis Theodoridis, "Generating Semantics-Based Trajectories of Moving Objects," International Workshop on Emerging Technologies for Geo-Based Applications, pp. 59-76, 2000.