

이동 객체의 불확실한 과거 및 미래의 위치 추정

(Estimation of Uncertain Past and Future Locations of Moving Objects)

안 윤 애[†] 류 근 호^{**}

(Yoon Ae Ahn) (Keun Ho Ryu)

요 약 데이터베이스를 이용하여 연속적인 이동 객체를 관리할 경우 매 시간마다 변경된 모든 위치 정보를 저장하는 것은 불가능하다. 그러므로, 일정한 비율의 시간 주기를 결정한 후 매 주기마다 이동 객체의 위치 정보를 이산적으로 저장하게 된다. 그러나 연속적인 모델의 이동 객체를 이산적인 형태로 관리하게 될 경우, 데이터베이스에 저장되지 않은 불확실한 과거 및 미래의 위치 정보에 대한 질의에 적절히 응답할 수 없는 문제점이 발생된다. 이러한 문제점을 해결하기 위해 이 논문에서는 동일한 데이터베이스에 저장된 이력정보를 이용한 이동 객체의 과거 및 미래의 위치 추정 방법과 알고리즘을 제안한다. 이 때 3차 스플라인 보간법을 적용하여 이동 객체의 과거 위치를 추정하고, 이력정보의 평균 이동 값을 구하여 미래의 위치를 추정한다. 결국, 가상 이동경로 및 위치 샘플을 이용한 위치 추정 실험을 통해 제안한 3차 스플라인 함수가 선형 함수보다 더 적은 오차를 발생시킴을 보였다.

키워드 : 이동 객체, 스플라인 보간법, 위치 추정, 위치 변화 함수

Abstract If continuous moving objects are managed by conventional database, it is not possible for them to store all position information changed over time in the database. Therefore, a time period of regular rate is determined and position information of moving objects are discretely stored in the system for every time period. However, if continuous moving objects are managed as discrete model, we will have problems which cannot properly answer to the query about uncertain past or future position information. To solve this problem, in this paper, we propose the method and algorithm which use the history information stored in the same database, to estimate the past or future location of moving objects. The cubic spline interpolation is used to estimate the past location and the mean movement value of the history information is used to predict the future location of moving objects. Finally, from the location estimation experimentation of using virtual trajectory and location sample, we proved that the proposed cubic spline function has less error than the linear function.

Key words : Moving Objects, Spline Interpolation, Location Estimation, Location Change Function

1. 서 론

이동 객체는 시간에 따라 객체의 위치 및 모양이 연속적으로 변경되는 시공간 데이터이다. 일반적으로 데이터베이스를 이용하여 연속적인 데이터를 관리할 경우, 매 초의 시간마다 변경된 모든 위치 정보를 저장하는 것은

불가능하므로 일정한 비율의 시간 주기를 결정한 후 매 시간 주기마다 객체의 위치 정보를 저장하게 된다. 이 때 데이터베이스에 저장되지 않은 불확실한 과거 및 미래의 위치 정보를 요구하는 질의에 대해 적절한 응답을 하지 못하는 문제가 발생된다. 이와 같은 문제는 이동 객체의 불확실한 위치 정보로 인해 발생되므로, 연속적인 모델의 이동 객체 데이터를 이산적인 형태로 데이터베이스에 저장 및 관리할 때 발생하는 이동 객체의 불확실성을 줄이고, 불확실한 위치 정보에 대한 질의에 적절히 응답할 수 있는 방법이 필요하다.

Sistla와 Wolfson[1,2,3]은 이동 객체의 현재 위치, 속도, 방향을 토대로 하여 미래의 위치를 예측하는 방법

· 이 연구는 한국과학재단 RRC(청주대 정보통신연구센터)의 지원으로 수행되었음.

† 학생회원 : 충북대학교 대학원 전자계산학과
yeahn@dbl.chungbuk.ac.kr

** 종신회원 : 충북대학교 전기전자 및 컴퓨터공학부 교수
khyu@dbl.chungbuk.ac.kr

논문접수 : 2001년 11월 8일

심사완료 : 2002년 9월 16일

을 제안하였지만, 이동 객체의 불확실한 과거의 이동 위치 추정 방법은 제시하지 않았다. Pfoser[4]는 이동 점 객체의 샘플되지 않은 불확실한 위치에 대해 선형 보간법을 이용한 방법을 제안하였다. 그러나 미래의 위치 추정 방법에 대한 언급이 전혀 없다. 또한 Pfoser[5]는 이동 객체의 시공간 변화에 대해 확률과 퍼지 집합 이론을 이용한 모델링 방법을 소개하였지만, 이동 객체의 표현 및 위치 추정에 대한 구체적인 방법을 제시하지 않았다. 지금까지의 연구들은 과거 또는 미래의 불확실한 위치 추정 중 한 가지에만 주로 초점을 맞추어 연구하였기 때문에, 동일한 데이터베이스에 저장된 이력 정보를 이용하여 동시에 과거와 미래의 이동 위치를 추정하는 방법을 제시하지 못하고 있다.

따라서, 이 논문에서는 동일한 이동 정보를 이용하여 이동 객체의 과거 및 미래의 위치를 추정하는 방법을 모두 제안한다. 먼저, 과거의 이동 정보는 일정한 샘플 구간에 의해 이력 데이터베이스에 저장된다. 샘플되지 않은 과거의 임의의 시점에 대한 위치 추정은 기존에 주로 사용되던 선형 보간법 대신 부분구간 3차 스플라인 보간법을 이용한다. 그리고 미래의 임의의 시점에 대한 위치 추정은 이력 데이터베이스에 저장된 과거 이동 정보의 평균값을 이용하여 추정한다.

그리고 제안 방법의 특성 분석을 위한 이동 경로 및 위치 샘플을 이용하여 위치 추정 실험을 한다. 이 실험을 통해 제안한 부분구간 3차 스플라인을 이용한 과거의 위치 추정 결과가 선형 보간법을 이용한 결과보다 오차가 적게 발생됨을 보인다. 아울러, 미래의 위치 추정에 사용되는 과거 이력 정보의 수를 변화시키며 실험한 후 결과에 대한 오차 값의 비교를 통해 그 특성을 검토한다.

이 논문의 내용 전개는 다음과 같다. 먼저, 2절에서는 지금까지 연구된 이동 객체의 개념 및 불확실성 처리 방법을 검토한다. 3절에서는 이동 객체의 불확실한 위치 추정 방법 및 알고리즘을 서술한다. 4절에서는 제안한 위치 추정 방법의 실험을 통해 추정된 위치의 오차를 분석한다. 마지막으로 5절에서는 결론 및 향후 연구방향을 제시한다.

2. 관련연구

이동 객체[3,6,7,8]는 시간에 따라 객체의 공간 정보가 연속적으로 변경되는 시공간 데이터로 이동 점과 이동 영역으로 구분된다. 이동 점(moving point)은 시간에 따라 객체의 위치가 변하는 것으로, 이동 점 객체의 예로는 사람이나 동물, 자동차, 비행기, 배 등이 있다. 이

동 영역(moving region)은 시간에 따라 객체의 위치뿐만 아니라 모양까지 변하는 것으로 한 국가의 행정 구역이나 폭풍의 영향권, 암세포의 상태 등을 예로 들 수 있다.

이와 같이 연속적으로 변하는 이동 객체를 모델링할 때는 항상 불확실성을 고려해야 한다. 왜냐하면, 아무리 정확한 측정 장비를 사용한다 할지라도 에러가 발생할 수 있으며, 객체의 위치 샘플을 얼마나 자주 획득하는가에 따라서 샘플되지 않은 위치의 삽입 시 발생하는 에러의 비율이 달라지기 때문이다. 이와 같은 이동 객체의 불확실성으로 인하여 데이터베이스의 모델링, 질의 처리, 인덱싱, 질의 결과의 부정확성 등과 같은 문제점들이 발생된다. 특히, 데이터의 불확실성으로 인해 발생하는 질의 결과의 부정확성은 사용자에게 잘못된 의사 결정 요인을 제공할 수도 있다[3,9,10,11].

이러한 문제점을 해결하기 위해 DOMINO[1,2,3] 프로젝트에서는 객체의 현재 위치, 속도, 방향을 토대로 하여 객체의 미래의 이동 위치를 예측하는 방법을 제안하였다. 이 연구에서는 위치 갱신 정책(position update policy)과 데이터베이스 표현에 포함된 위치의 부정확성을 언급하고 있으며, 이동 객체의 현재와 미래의 위치를 추적하는 질의를 취급하는 방법을 제안하고 있다. 불확실성 및 부정확성을 관리하기 위한 몇 가지 방안으로 데이터 모델의 확장, 질의어의 확장, 확률을 이용한 불확실한 질의에 대한 응답, 데이터베이스의 갱신 비용과 불확실성 비용의 절충안 등을 제시하고 있다. 그러나 이 연구에서는 이동 객체의 불확실한 과거의 이동 위치에 대한 이력정보를 데이터베이스에 저장하지 않고 있으며, 과거의 위치를 추정하는 방법이 제시되지 않고 있다.

CHOROCHRONOS[4,5] 프로젝트에서 Pfoser는 관계형 데이터베이스를 이용한 이동 객체의 표현 방법 및 여러 정보를 이용한 불확실성의 정량화 방법을 제안하고 있다. 곡선으로 표현되는 이동 객체의 연속적인 움직임 이산적인 형태의 폴리라인으로 표현할 때 발생하는 이동 객체의 불확실성 발생 요인을 측정 에러와 샘플링 에러로 구분하고 있다. 그리고 시공간 데이터와 여러 정보를 관계형 데이터베이스 스키마에 통합하는 방법을 제안하고 있으며, 이동 점 객체의 위치 표현에서 샘플링되지 않은 불확실한 위치에 대해 선형 보간법을 사용하여 정량화하는 방법을 제안하고 있다. 그러나 구체적인 실험을 통한 정확성의 검증이 결여된 방법론의 제시이며, 미래의 위치 추정 방법에 대한 언급이 전혀 없다.

그리고 Pfoser는 [5]에서 시공간 시나리오를 나타내

는 이동 객체의 변화에 대한 몇 가지 예를 들어 확률과 퍼지 집합 이론을 이용하여 불확실한 변화를 모델링하는 방법을 소개하고 있다. 이 연구에서는 시공간 불확실성의 요소를 공간 객체, 시간 점, 시간 구간으로 구분하고 있으며, 서로 다른 네 가지 종류의 시공간 시나리오와 변화에 대해 퍼지 집합 또는 확률을 이용한 변화 함수의 불확실성 처리 방법을 제안하고 있다. 그러나 데이터베이스를 이용한 이동 객체의 표현 및 불확실한 위치를 추정하는 구체적인 방법에 대한 제시 및 실험이 결여되었다.

3. 이동 객체의 과거 및 미래의 위치 추정

시공간 응용에서는 시간에 따른 공간 객체들의 관계 및 속성들에 많은 관심을 가지며, 현실적으로 시간에 따른 진화(evolution)와 변화(change) 정보를 주로 기록한다. 이 중에서도 시간에 따른 공간 객체의 변화는 시공간 데이터베이스에서 가장 중요한 개념으로 취급되고 있으며, 이동 차량, 사람, 군부대 등과 같은 이동 객체의 응용에서는 시간에 따른 객체의 위치 변화를 중점적으로 다룬다. 이동 객체의 변화 시나리오는 시공간 불확실성을 평가하는 기초 도구로 사용된다[5]. 이와 같은 이동 객체의 위치 변화는 이산적 또는 연속적으로 발생할 수 있으며, 시간 점(time points) 또는 시간 구간(time periods)으로 기록될 수 있다[4,7].

3.1 위치 변화 시나리오

이 논문에서는 이동 객체의 불확실한 위치정보 처리를 위해 그림 1과 같이 연속적으로 변화하는 이동 객체의 위치 변화 시나리오를 대상으로 과거 및 미래의 위치를 추정한다. 또한, 이동 객체는 이동 점만을 대상으로 함을 가정한다.

그림 1의 시나리오는 연속적으로 이동하는 객체를 나타내고 있으며, 객체의 이동 위치는 이산적인 시점에서 샘플 됨을 가정한다. 이동 객체는 이차원 공간상의 x, y

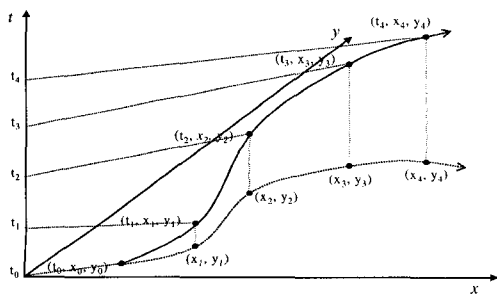


그림 1 이동 객체의 연속적인 위치 변화

좌표를 시간을 나타내는 t 축으로 확장하여 3차원으로 구성한다. t 축이 가질 수 있는 시간 값은 과거, 현재, 미래가 모두 가능하다. 그러나, 실제 데이터베이스에 저장되는 값은 과거와 현재 값이고, 미래의 값은 위치 예측 함수의 연산에 의해 그 결과를 얻는다.

3.2 데이터 표현

이 논문에서는 단순히 일반적인 관계형 데이터베이스를 기반으로 하여 이동 객체의 위치 정보를 저장한다. 그리고 저장된 데이터를 이용하여 시간에 따른 위치 변화 함수 및 불확실한 위치정보를 추정한다. 위치 샘플이 획득된 이동 객체를 저장하기 위한 데이터베이스는 모두 두 개의 릴레이션으로 구성되며 다음과 같은 스키마를 가진다.

OBJECT(Oid, Code, Name, Type, Etc.)
 HISTORY(Code, Time, X, Y)

객체정보 릴레이션 OBJECT는 이동 객체에 대한 비공간 속성 정보를 저장한다.

표 1 OBJECT 릴레이션

| OBJECT | Oid | Code | Name | Type | Etc |
|--------|------|------|------|-------|-----|
| | 1234 | c1 | Oa | truck | ... |
| | 2345 | c2 | Ob | taxi | ... |
| | 3456 | c3 | Oc | bus | ... |
| | ⋮ | ⋮ | ⋮ | ⋮ | ... |

표 1의 OBJECT 릴레이션에서 Oid는 이동 객체의 식별자로서 키 값이 된다. Code, Name, Type, Etc 속성은 각각 객체의 코드, 이름, 종류, 기타 비공간 정보를 나타낸다. 다음의 이동정보 릴레이션 HISTORY는 샘플된 객체의 시간 및 위치에 관한 이력 정보를 저장한다.

표 2의 HISTORY 릴레이션에서 Code는 키 값을 나타내며, Time, X, Y는 샘플된 시점과 위치 좌표 값을 나타낸다.

표 2 HISTORY 릴레이션

| HISTORY | Code | Time | X | Y |
|---------|------|------|-----|-----|
| | c1 | 0 | 242 | 262 |
| | c1 | 5 | 250 | 266 |
| | c1 | 10 | 256 | 268 |
| | c1 | 15 | 260 | 272 |
| | c2 | 0 | 150 | 220 |
| | c2 | 5 | 153 | 224 |
| | ⋮ | ⋮ | ⋮ | ⋮ |

3.3 과거의 위치 추정

이동 객체의 위치 정보는 일정한 시간 주기마다 데이터베이스에 저장된다. 저장되지 않은 임의의 과거 시점 t_k 에 대한 객체의 위치 정보에 관한 질의 결과를 얻기 위해서 시간에 종속적인 위치 변화 함수를 사용한다[12]. 지금까지 이동 객체의 위치 변화 함수에 주로 사용된 방법은 선형 보간법[4,13,18]이다. 선형 보간법은 두 점을 이용한 1차 함수를 구하여 연속적인 위치 변화를 표현하므로, 객체의 이동은 모든 구간에서 직선의 형태로 나타난다. 따라서 선형 보간법을 이용할 경우에는 이 논문에서 가정한 그림 1의 위치 변화 시나리오 같은 완전한 곡선 형태의 이동을 표현할 수 없다. 따라서 이 논문에서는 다음과 같이 부분구간 3차 스플라인 보간법을 이용한 위치 변화 함수를 제안한다.

3.3.1 부분구간 3차 스플라인 보간법을 이용한 위치 변화 함수

완만한 곡선의 형태로 움직이는 위치 변화 시나리오의 불확실한 과거 위치를 추정하기 위해 다음과 같이 부분구간 3차 스플라인 보간법을 이용한 위치 변화 함수를 사용한다. 이 함수는 과거의 위치 추정에 사용되며, 다음의 정의 1, 2, 3을 이용한다.

[정의 1] (위치 변화 함수) 위치 변화 함수는 이동 객체의 연속적인 위치 변화를 모델링하는 것으로, 임의의 시간 구간 $T_i = [t_i, t_{i+1}]$ 내의 모든 객체의 위치를 반환할 수 있는 함수이다[12]. 위치 변화 함수는 $PCF : t_k \rightarrow (x_k, y_k)$ 형태를 띠며, 임의의 시간 t_k 를 입력 값으로 받아서 그 결과로 (x_k, y_k) 의 위치 좌표를 반환한다. 이 때 t_k 의 범위는 $t_0 \leq t_k \leq now$ 이다. t_0 는 객체정보 릴레이션 OBJECT에 저장된 최초의 시점이고, now 는 이동정보 릴레이션 HISTORY에 저장된 가장 최근의 시점이다. □

[정의 2] (3차 스플라인 보간법) $\{(x_i, y_i)\}_{i=0}^N$ 가 $N+1$ 개의 점으로 이루어진 집합이라 하자. 여기서 $a = x_0 < x_1 < \dots < x_N = b$ 일 때, 다음 성질을 만족하는 $s_{i,0}, s_{i,1}, s_{i,2}, s_{i,3}$ 을 계수로 하는 N 개의 3차 다항식 $S_i(x)$ 가 존재하면, 함수 $S(x)$ 를 3차 스플라인(cubic spline)이라고 한다[14].

성질 I.

$$S(x) = S_i(x) = s_{i,0} + s_{i,1}(x-x_i) + s_{i,2}(x-x_i)^2 + s_{i,3}(x-x_i)^3, x \in [x_i, x_{i+1}], i=0,1,\dots,N-1$$

성질 II. $S(x_i) = y_i, i=0,1,\dots,N$

성질 III. $S_i'(x_{i+1}) = S_{i+1}'(x_{i+1}), i=0,1,\dots,N-2$

성질 IV. $S_i''(x_{i+1}) = S_{i+1}''(x_{i+1}), i=0,1,\dots,N-2$

성질 V. $S_i'''(x_{i+1}) = S_{i+1}'''(x_{i+1}), i=0,1,\dots,N-2$ □

3차 스플라인 보간법은 데이터 점들의 집합을 다항식 곡선으로 나타낸 것으로, CAD/CAM, 컴퓨터 그래픽스 시스템 등에서 주로 사용한다. 이것의 목적은 주어진 점들을 지나면서 오차에 영향을 받지 않는 유연한 곡선을 그리는 것이다. 3차 스플라인 보간법에 대한 구체적인 정리와 증명은 [14,15,16]에 제시되어 있다.

다음 그림 2는 부분구간 3차 스플라인 보간법을 나타낸 것이다. $N+1$ 개의 점으로 이루어진 집합 $\{(x_i, y_i)\}_{i=0}^N$ 이 존재할 때 N 개의 부분구간 3차 다항식을 구함으로써, 연속적으로 존재하는 모든 구간의 좌표 값을 구할 수 있음을 보이고 있다. 임의의 x 좌표 x_k 가 $x_i < x_k < x_{i+1}$ 과 같은 범위에 존재할 경우 y 좌표를 구하는 부분구간 3차 스플라인 보간 다항식은 $(x_{i-1}, y_{i-1}), (x_i, y_i), (x_{i+1}, y_{i+1}), (x_{i+2}, y_{i+2})$ 의 4점을 이용하여 구한다.

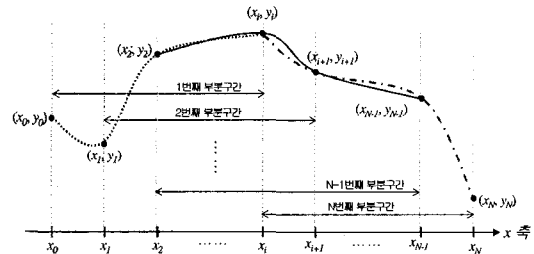


그림 2 부분구간 3차 스플라인 보간법

정의 2를 이용하여 임의의 이동 객체 O_A 에 대한 위치 변화 함수 PCF 를 정의하기 위해 다음과 같이 이동 객체의 이력 집합을 정의한다. 이동 객체는 (x, y) 의 2차원 공간 정보에 1차원 시간 정보를 더한 3차원 집합의 형태로 데이터베이스에 저장된다. 그러나 위치 변화 함수는 3차원 집합을 대상으로 하지 않고, 시간 t 에 대해 각각 x 축과 y 축으로 분리한 2차원 집합을 대상으로 한다.

[정의 3] (이동 객체의 이력 집합) 임의의 이동 객체 O_A 의 모든 이력 집합을 P 라 하자. 이 때, 3차원 이력 집합 $P = \{p_0(t_0, x_0, y_0), p_1(t_1, x_1, y_1), \dots, p_n(t_n, x_n, y_n)\}$ 이 된다. P 에서 $p_0(t_0, x_0, y_0)$ 는 HISTORY 릴레이션에 저장된 최초의 시점 t_0 에 대한 위치 좌표 (x_0, y_0) 를 의미한다. 그리고 $p_n(t_n, x_n, y_n)$ 은 가장 최근에 저장된 시점 t_n 에 대한 위치 좌표 (x_n, y_n) 를 의미한다. 3차원 이력 집

합 P 를 시점 t 에 대해 각각 x 와 y 에 대해 2차원 집합으로 분해한다. 이 때, x 축 이력 집합 $P_x = \{p_{x_0}(t_{x_0}, x_0), p_{x_1}(t_{x_1}, x_1), \dots, p_{x_n}(t_{x_n}, x_n)\}$, y 축 이력 집합 $P_y = \{p_{y_0}(t_{y_0}, y_0), p_{y_1}(t_{y_1}, y_1), \dots, p_{y_n}(t_{y_n}, y_n)\}$ 이라고 정의한다. □

정의 1, 2, 3을 바탕으로 과거의 임의의 시점 t_k 에 대한 x, y 좌표를 구하는 부분구간 3차 다항식은 2차원 이력 집합 P_x 와 P_y 를 이용하여 $x = S_i(t_k)$ 와 $y = S_i(t_k)$ 두 개의 부분구간 3차 스플라인 함수를 구하는 것이다.

먼저, 시간의 변화에 따른 x 축 이력 집합 P_x 는 그림 3과 같이 표현할 수 있다. 이 그래프는 시간 t_x 에 따른 x 의 변화를 나타낸 것으로, (t_x, x_i) 는 i 번째의 시간 값과 x 좌표 값을 의미한다.

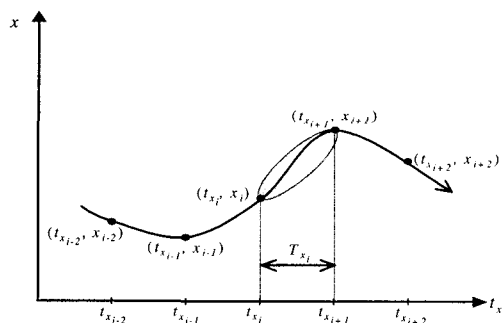


그림 3 시간에 따른 이동 객체의 x 좌표 변화

그림 3에서 임의의 시간 구간 $T_{x_i} = [t_{x_i}, t_{x_{i+1}}]$ 에 대해 $t_{x_i} < t_k < t_{x_{i+1}}$ 의 조건을 만족하는 과거의 임의의 시점 t_k 에 대한 x 좌표를 구하는 함수는 $S_i(t_k)$ 이다. 함수 $S_i(t_k)$ 는 4점의 좌표 $(t_{x_{i-1}}, x_{i-1}), (t_{x_i}, x_i), (t_{x_{i+1}}, x_{i+1}), (t_{x_{i+2}}, x_{i+2})$ 을 이용하여 다음 식 (1)과 같이 구한다.

$$x = S_i(t_k) = ((s_{i,3}w + s_{i,2})w + s_{i,1})w + s_{i,0}$$

$$w = t_k - t_{x_i}, \quad s_{i,0} = x_i, \quad s_{i,1} = d_i - \frac{h_i(2m_i + m_{i+1})}{6}$$

$$s_{i,2} = \frac{m_i}{2}, \quad s_{i,3} = \frac{m_{i+1} - m_i}{6h_i}$$

$$h_i = t_{x_{i+1}} - t_{x_i}, \quad d_i = \frac{x_{i+1} - x_i}{h_i} \quad \text{식 (1)}$$

시간의 변화에 따른 y 축 이력 집합 P_y 는 다음 그림 4와 같이 표현되며, 시간 t_y 에 따른 y 의 변화를 그래프 형태로 나타낸 것이다. 여기에서 (t_y, y_i) 는 i 번째의 시간 값과 y 좌표 값을 의미한다.

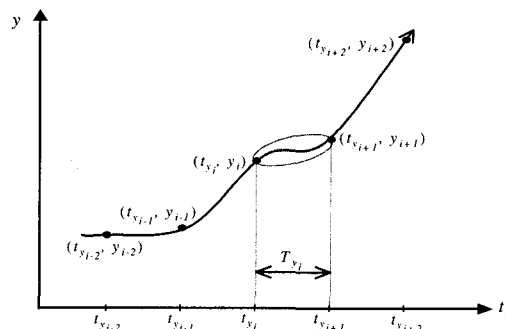


그림 4 시간에 따른 이동 객체의 y 좌표 변화

그림 4는 3차원 이동 경로를 y 에 대한 2차원으로 표현한 것이다. 임의의 시간 구간 $T_{y_i} = [t_{y_i}, t_{y_{i+1}}]$ 에 대해 $t_{y_i} < t_k < t_{y_{i+1}}$ 의 조건을 만족하는 과거의 임의의 시점 t_k 에 대한 y 좌표를 구하는 함수는 $S_i(t_k)$ 이다. 함수 $S_i(t_k)$ 는 4개의 점 $(t_{y_{i-1}}, y_{i-1}), (t_{y_i}, y_i), (t_{y_{i+1}}, y_{i+1}), (t_{y_{i+2}}, y_{i+2})$ 을 이용하여 다음 식 (2)와 같이 구한다.

$$y = S_i(t_k) = ((s_{i,3}w + s_{i,2})w + s_{i,1})w + s_{i,0}$$

$$w = t_k - t_{y_i}, \quad s_{i,0} = y_i, \quad s_{i,1} = d_i - \frac{h_i(2m_i + m_{i+1})}{6}$$

$$s_{i,2} = \frac{m_i}{2}, \quad s_{i,3} = \frac{m_{i+1} - m_i}{6h_i}$$

$$h_i = t_{y_{i+1}} - t_{y_i}, \quad d_i = \frac{y_{i+1} - y_i}{h_i} \quad \text{식 (2)}$$

스플라인 계수를 구하는 풀이 과정은 [14,15,16]에 좀 더 자세하게 제시되어 있으며, 알고리즘은 다음 절에서 소개한다.

3.3.2 과거의 위치 추정 알고리즘

데이터베이스의 이력 릴레이션에 저장되지 않은 임의의 시점 t_k 에 대한 위치 정보 추정 알고리즘은 그림 5와 같다. 이 알고리즘은 3.2절의 OBJECT 릴레이션과 HISTORY 릴레이션을 대상으로 하며, 4점을 이용한 부분구간 3차 스플라인 보간 다항식을 이용한 위치 변화 함수를 사용한다.

그림 5의 위치 변화 함수 알고리즘 PositionChange Function은 임의의 이동 객체 OA의 식별자 oid_O와 과거의 임의의 시점 t_k 를 입력받은 후, t_k 시점의 (x, y) 좌표인 x_t_k 와 y_t_k 를 결과로 반환하는 함수이다. 먼저, 입력된 객체의 식별자 oid_O가 OBJECT 릴레이션에 저장된 Oid의 인스턴스인지를 검사한다. 만약, 입력된 객체의 식별자가 객체정보 릴레이션 OBJECT에 포함되지 않으면 결과변수 x_t_k 와 y_t_k 에 오류 값을 할당한 후 함수를 종료한다. 그렇지 않은 경우에는 HISTORY 릴레이션에서 입력된 객체에 대해 t_k 시점

```

Algorithm PositionChangeFunction(oid, O, t, k)
입력 : oid, O(객체 O의 식별자), t, k(과거의 임의의 시점)
출력 : x_t,k(시점 t_k의 x 좌표 값), y_t,k(시점 t_k의 y 좌표 값)
OBJECT : 객체정보 릴레이션, HISTORY : 이력정보 릴레이션
Begin
  If (OBJECT 릴레이션에서 Oid = oid, O인 객체가 존재) Then
    임시변수 code에 객체의 Code값을 할당
    If (HISTORY 릴레이션에서 Code = code and Time = t, k 튜플 검색) Then
      x_t,k에 X값을 할당, y_t,k에 Y값을 할당
    Else
      HISTORY에서 t, k 시점 바로 이전의 2개의 튜플과 바로 이후의 2개의 튜플 검색
      Time1, Time2, Time3, Time4에 순서대로 4개 튜플의 Time값을 할당
      X1, X2, X3, X4에 순서대로 4개 튜플의 X값을 할당
      Y1, Y2, Y3, Y4에 순서대로 4개 튜플의 Y값을 할당
      x_t,k ← SplineInterpolation(Time, X, t, k) 실행 후 x_t,k 값을 구함
      y_t,k ← SplineInterpolation(Time, Y, t, k) 실행 후 y_t,k 값을 구함
    Else 결과변수 x_t,k와 y_t,k에 오류 값 할당
  결과 값 x_t,k와 y_t,k를 반환
End
  
```

그림 5 위치 변화 함수를 이용한 과거의 위치 추정 알고리즘

```

Algorithm SplineInterpolation(time, loc, t, k)
입력 : ① 3차 스플라인 다항식을 구하는데 사용되는 좌표 쌍의 집합
      (time0, loc0), (time1, loc1), (time2, loc2), (time3, loc3) 4점의 좌표 집합을 입력받음
      ② 과거의 임의의 시점 t, k
출력 : 임의의 시점 t, k의 x 또는 y 위치 값(변수 : Coord_XorY)
N : 입력 좌표 쌍의 수(N = 4로 초기화, 4점을 이용한 3차 스플라인 보간 다항식을 구함)
Begin
  Do i = 0, N-2
    hi = timei+1 - timei
  End Do
  Do i = 0, N-3
    m1i = 6 * ((loci+2 - loci+1) / hi+1 - (loci+1 - loci) / hi)
  End Do
  m10 = h1
  m20 = 3 * (h0 + 2 * h1)
  m21 = 3 * (h2 + 2 * h1)
  m30 = h1
  Do i = 1, N-3
    m2i = m2i - ((m1i / m2i-1) * m3i-1)
    m1i = m1i - ((m1i / m2i-1) * m1i-1)
  End Do
  s1 = m10 / m20, s0 = s1, s2 = s2
  Do i = 0, N-2
    ai = (si+1 - si) / 6 * hi
    bi = si / 2
    ci = (loci+1 - loci) / hi - (2 * hi * si + hi * si+1) / 6
    di = loci
  End Do
  If (time0 < t, k < time3) Then
    Coord_XorY = di + (t, k - time0) * (ci + (t, k - time0) * (bi + ai * (t, k - time0)))
  End Do
  시점 t, k 에 대해 추정된 위치 값 Coord_XorY를 반환
End
  
```

그림 6 점을 이용한 스플라인 보간 다항식 알고리즘

의 위치 좌표를 검색한다. t, k 시점의 위치 값이 이동정보 릴레이션에 존재하면 검색된 값을 결과로 반환한 후 함수를 종료하고, 그렇지 않으면 부분구간 3차 스플라인 함수를 호출하여 x_t,k와 y_t,k를 구한다.

이 때 SplineInterpolation 함수 호출에 사용되는 입력 값 Time은 스플라인 함수를 구하기 위한 4점의 시간 값이고, X와 Y는 스플라인 함수를 구하기 위한 4점의 x좌표와 y좌표 값이다. 4점의 위치는 입력된 시점 t, k 바로 이전의 2개의 이력 튜플과 바로 이후의 2개의 이력 튜플을 의미한다. 스플라인 함수를 처리하는 알고리즘은 그림 6과 같다.

그림 6의 SplineInterpolation 함수에는 3차 스플라인 다항식을 구하는데 필요한 4점의 좌표 쌍의 집합과 과거의 임의의 시점 t, k가 입력된다. 4점의 좌표 쌍은 (time₀, loc₀), (time₁, loc₁), (time₂, loc₂), (time₃, loc₃)의

형태로 구성된다. 여기에서 time은 이력정보 릴레이션에 저장된 시간 값이고, loc은 time 시점에 이동 객체가 위치하는 x 또는 y의 좌표 값을 의미한다. 이 함수의 결과 값은 Coord_XorY의 변수에 저장된 값으로, 입력 좌표 쌍으로부터 생성된 스플라인 함수를 이용한 시점 t, k의 값이다.

SplineInterpolation 함수의 알고리즘은 3차 스플라인 보간 다항식을 구하는 식을 그대로 전개한 것이다. 먼저, 스플라인 다항식의 4개의 계수 a_i, b_i, c_i, d_i를 계산하고, 구해진 계수들과 입력된 4점의 좌표 쌍을 이용하여 시점 t, k의 x 또는 y 값을 구한 후 그 결과 값을 반환한다.

3.4 미래의 위치정보 추정

이동 객체를 이용한 대부분의 응용[3,17,18]은 차량의 이동 궤적 관리 및 이동 경로의 예측에 초점을 맞추고

있다. 이동 궤적의 관리는 데이터베이스에 저장된 이력 정보를 이용하여 질의 결과를 제공할 수 있지만, 향후 이동 경로의 예측은 특정 응용에 적합한 여러 가지 상황을 고려해야 한다. 이동 객체의 미래 위치 추정을 위해 wolfson[1,2,3,17]은 이동 객체의 현재 위치, 속도를 토대로 하여 미래의 위치를 제공하는 방법을 제안하였다. 그러나 이 논문에서는 속도와 방향 정보를 데이터베이스에 저장하지 않고 이력 릴레이션에 저장된 과거의 이동 정보만을 이용한 위치 추정 방법을 제안한다. 이동 객체의 미래의 이동 위치는 위치 예측 함수(position prediction function)에 의해 처리되며, 위치 예측 함수를 이용한 위치 추정을 위해 다음과 같은 정의를 이용한다.

[정의 4] (위치 예측 함수) 위치 예측 함수는 $PPF : t_f \rightarrow (x_{t_f}, y_{t_f})$ 형태를 띠며, 미래의 임의의 시점 t_f 를 입력 값으로 받아 그 결과로 (x_{t_f}, y_{t_f}) 의 위치 좌표를 반환한다. 이 때 시점 t_f 의 범위는 $now < t_f$ 이다. now 는 이동정보 릴레이션에 저장된 가장 최근의 시점 이므로, t_f 는 항상 미래의 임의의 특정 시점을 나타낸다. 위치 예측 함수에 사용되는 표기들은 다음과 같다.

- t_f : $PPF(t_f)$ 의 입력 값, 미래의 특정 시점을 의미
- d_{mx} : x 축으로의 평균 이동거리
- d_{my} : y 축으로의 평균 이동거리
- t_{mf} : 현재부터 미래의 특정 시점까지의 이동시간
- d_{tx} : x 축으로의 총 이동거리
- d_{ty} : y 축으로의 총 이동거리

- t_{tx} : x 축으로의 총 이동시간
- t_{ty} : y 축으로의 총 이동시간
- t_{x_n} : x 축 이력 집합의 가장 최근의 시점
- t_{y_n} : y 축 이력 집합의 가장 최근의 시점
- x_n : 가장 최근 시점의 x 좌표
- y_n : 가장 최근 시점의 y 좌표

미래의 임의의 시점 t_f 에 대한 x, y 좌표를 구하는 위치 예측 함수는 정의 3의 x 축 이력 집합 P_x 와 y 축 이력 집합 P_y 를 대상으로 정의 4와 같은 표기들을 사용한다. 이동 객체 O_A 에 대한 미래의 위치 추정은 x 축과 y 축 각각에 대해 $x = PPF(t_f)$ 와 $y = PPF(t_f)$ 두 개의 함수를 이용하여 구한다. 먼저, 미래의 특정 시점 t_f 에 대한 x 좌표 값은 이력 집합 P_x 를 대상으로 다음의 식 (3)과 같이 구한다.

$$\begin{aligned} x &= (d_{mx} \times t_{mf}) + x_n \\ &= ((d_{tx} \div t_{tx}) \times (t_f - t_{x_n})) + x_n \\ &= \left(\sum_{i=2}^n |x_i - x_{i-1}| \div \sum_{i=2}^n (t_{x_i} - t_{x_{i-1}}) \right) \\ &\quad \times (t_f - t_{x_n}) + x_n \end{aligned} \quad \text{식 (3)}$$

미래의 특정 시점 t_f 에 대한 y 좌표 값은 이력 집합 P_y 를 대상으로 하여 다음의 식 (4)를 이용하여 구한다.

$$\begin{aligned} y &= (d_{my} \times t_{mf}) + y_n \\ &= ((d_{ty} \div t_{ty}) \times (t_f - t_{y_n})) + y_n \\ &= \left(\sum_{i=2}^n |y_i - y_{i-1}| \div \sum_{i=2}^n (t_{y_i} - t_{y_{i-1}}) \right) \\ &\quad \times (t_f - t_{y_n}) + y_n \end{aligned} \quad \text{식 (4)}$$

```

Algorithm PositionPredictionFunction(oid_O, t_f)
입력 : oid_O(객체 O_A의 식별자), t_f(미래의 임의의 시점)
출력 : x_t_f(시점 t_f의 x 좌표값), y_t_f(시점 t_f의 y 좌표값)
OBJECT : 객체정보 릴레이션, IHISTORY : 이동정보 릴레이션
N : 미래의 위치 좌표 추정에 사용될 최근 이동정보의 튜플 수
Begin
  If (OBJECT 릴레이션에서 Oid = oid_O인 객체가 존재) Then
    임시변수 code에 객체의 Code값을 할당
    If (IHISTORY 릴레이션에서 Code = code and Time = t_f 튜플 검색) Then
      x_t_f에 X값을 할당, y_t_f에 Y값을 할당
    Else
      current ← IHISTORY에서 가장 마지막에 저장된 튜플의 time 값
      current_x ← current 시점의 x 좌표 값
      current_y ← current 시점의 y 좌표 값
      IHISTORY에서 current 이전의 N개의 튜플을 검색
      검색된 N개의 튜플의 time, x, y 값을 t_i, x_i, y_i 배열에 저장
      d_Lx, d_Ly, L_Lx에 초기치 0을 할당
      DO i=0, N-2
        d_x = x_{i+1} - x_i //현재 튜플의 x축 이동거리
        d_Lx = d_Lx + d_x //x축 이동거리의 합
        d_y = y_{i+1} - y_i //현재 튜플의 y축 이동거리
        d_Ly = d_Ly + d_y //y축 이동거리의 합
        L_x = t_{i+1} - t_i //현재 튜플의 이동시간
        L_Lx = L_Lx + L_x //이동시간의 합
      End DO
      d_m_x = d_Lx / L_Lx //x축 평균 이동거리
      d_m_y = d_Ly / L_Ly //y축 평균 이동거리
      t_m_f = t_f - current //향후 이동시간
      x_t_f = (d_m_x * t_m_f) + current_x //미래의 x좌표
      y_t_f = (d_m_y * t_m_f) + current_y //미래의 y좌표
    Else 결과변수 x_t_f와 y_t_f에 오류 값 할당
  End
  결과 값 x_t_f와 y_t_f를 반환
  
```

그림 7 미래의 위치 추정 알고리즘

식 (3)과 (4)를 이용한 미래의 위치 추정 알고리즘은 3.2절의 객체정보 릴레이션 OBJECT와 이동정보 릴레이션 HISTORY를 대상으로 한다.

그림 7의 알고리즘 PositionPredictionFunction의 입력 값은 임의의 이동 객체 O_A 의 식별자인 oid_O 와 미래의 임의의 시점 t_f 이다. 출력 값은 t_f 시점의 (x, y) 좌표인 x_{t_f} 와 y_{t_f} 이다. 먼저, 입력된 객체의 식별자 oid_O 가 OBJECT 릴레이션에 저장된 Oid의 인스턴스 인지를 검사한다. 만약, 입력된 객체의 식별자가 객체정보 릴레이션 OBJECT에 포함되지 않으면 결과변수 x_{t_f} 와 y_{t_f} 에 오류 값을 할당한 후 함수를 종료한다. 그렇지 않은 경우에는 HISTORY 릴레이션에서 입력된 객체에 대해 t_f 시점의 위치 좌표를 검색한다. t_f 시점의 위치 값이 이동정보 릴레이션에 존재하면 검색된 값을 결과로 반환한 후 함수를 종료하고, 그렇지 않으면 미래의 위치 추정 함수를 이용하여 x_{t_f} 와 y_{t_f} 를 구한 후 결과 값을 반환한다. 미래 위치 추정 알고리즘의 Do 루프 문에 사용된 변수 N은 미래의 위치 추정에 사용될 과거 이력 정보의 수를 나타낸다. N의 값은 2이상 어떤 수라도 가능하다.

4. 실험

제한한 위치 추정 알고리즘의 정확성 분석을 위해 데이터 샘플을 만들어 실험하였다. 실험 환경은 Window 2000에서 Java 언어와 Oracle DBMS를 사용하였으며, 데이터베이스 연동을 위해 JDBC를 이용하였다. 실험 대상이 되는 이동 객체는 이동 점으로 간주하며, 도로에서 이동하는 차량으로 한정한다. 그리고 이동 차량은 거의 일정한 속도로 이동하고 있음을 가정한다.

위치 추정 실험 시 추정된 위치와 실제 위치 좌표와의 오차 값을 계산하기 위해서는 실험 대상이 되는 이동 경로의 완전한 이력 정보가 필요하다. 따라서 이 실험에서는 표 3, 4와 같이 고속도로 상의 임의의 두 구간에 대한 완전한 이동 경로를 선정하여 각각의 이동 시점과 위치 좌표에 관한 정보를 데이터베이스에 미리 저장하였다. 구간 1은 고속도로에서 “청원→신탄진→회덕→서대전” 방향으로 이동하는 경로이고, 구간 2는 “청원→신탄진→회덕→옥천” 방향으로 이동하는 경로이다.

그림 8은 구간 1의 완전한 이동 경로를 나타낸 화면이다. 표 3은 구간 1의 완전한 이동 정보가 저장된 이력 릴레이션의 일부를 나타낸 것으로, 이력 릴레이션의 전체 튜플은 200개이다. 사용된 좌표 값은 Java 프로그래밍 언어의 좌표 체계를 그대로 사용하였으며, 좌표의 단

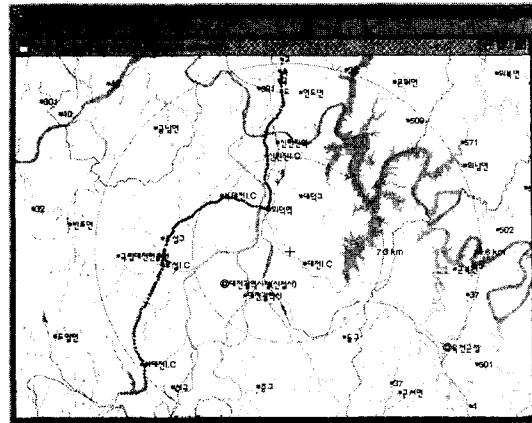


그림 8 구간 1의 이동 경로

표 3 구간 1의 이동정보 릴레이션

| Code | Time | X | Y |
|------|------|-------|-------|
| c1 | 0 | 380.0 | 22.0 |
| c1 | 1 | 379.0 | 37.0 |
| c1 | 2 | 381.0 | 56.0 |
| c1 | 3 | 381.0 | 73.0 |
| c1 | 4 | 369.0 | 91.0 |
| c1 | 5 | 359.0 | 106.0 |
| : | : | : | : |
| c1 | 80 | 294.0 | 209.0 |
| c1 | 81 | 281.0 | 215.0 |
| c1 | 82 | 269.0 | 221.0 |
| c1 | 83 | 257.0 | 229.0 |
| c1 | 84 | 246.0 | 237.0 |
| c1 | 85 | 232.0 | 243.0 |
| : | : | : | : |
| c1 | 150 | 168.0 | 405.0 |
| c1 | 151 | 170.0 | 429.0 |
| c1 | 152 | 180.0 | 449.0 |
| c1 | 153 | 183.0 | 473.0 |
| c1 | 154 | 178.0 | 493.0 |
| c1 | 155 | 157.0 | 496.0 |
| : | : | : | : |

위는 픽셀이다. 사용된 지도 화면은 실제 지도의 이미지를 사용하였으며, 저장된 위치 샘플은 1 픽셀당 실제 거리로 약 30m 정도에 해당된다.

그림 9는 구간 2의 완전한 이동경로를 나타낸 화면이며, 표 4는 구간 2의 완전한 이동정보가 저장된 이력 릴레이션의 일부이다. 전체 이력 릴레이션은 구간 1과 마찬가지로 200개의 튜플로 구성된다.

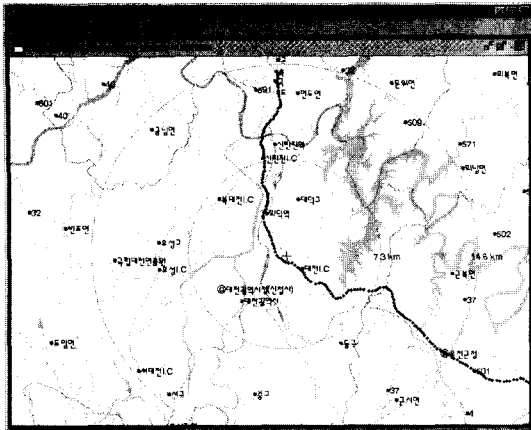


그림 9 구간 2의 이동 경로

표 4 구간 2의 이동정보 릴레이션

| Code | Time | X | Y |
|------|------|-------|-------|
| c2 | 0 | 380.0 | 22.0 |
| c2 | 1 | 379.0 | 37.0 |
| c2 | 2 | 381.0 | 56.0 |
| c2 | 3 | 380.0 | 73.0 |
| c2 | 4 | 369.0 | 91.0 |
| c2 | 5 | 359.0 | 106.0 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| c2 | 80 | 389.0 | 297.0 |
| c2 | 81 | 409.0 | 306.0 |
| c2 | 82 | 421.0 | 322.0 |
| c2 | 83 | 445.0 | 329.0 |
| c2 | 84 | 457.0 | 346.0 |
| c2 | 85 | 476.0 | 341.0 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| c2 | 150 | 622.0 | 432.0 |
| c2 | 151 | 642.0 | 444.0 |
| c2 | 152 | 664.0 | 457.0 |
| c2 | 153 | 684.0 | 468.0 |
| c2 | 154 | 709.0 | 469.0 |
| c2 | 155 | 733.0 | 470.0 |
| ⋮ | ⋮ | ⋮ | ⋮ |

4.1 과거의 위치 추정

표 3, 4의 구간 1과 2의 이동 정보 릴레이션을 대상으로 3절에서 제안한 부분구간 3차 스플라인 함수를 이용한 임의의 과거 시점에 대한 위치 추정 및 오차 값의 비교를 위해 두 구간의 이동 경로에 대해 위치 샘플을 획득하였다. 샘플링은 사용자 인터페이스의 지도 화면에서 해당 구간의 이동 경로를 따라서 일정한 간격으로 마우스를 클릭하여 해당 좌표 값이 이력 정보 릴레이션에 자동으로 저장되도록 하였다. 또한, 이력 릴레이션에 저장된 이동 시간은 샘플링 간격에 비례하도록 저장하였다. 샘플 간격은 약 0.2km ~ 1.6km의 거리 간격으로 8회에 걸쳐 샘플링 하였으며, 각각 서로 다른 릴레이션에 저장하였다. 실험은 선형 함수를 이용한 위치 추정 방법과 3차 스플라인 보간 다항식을 이용한 위치 추정 방법을 각각 따로 수행하여 그 결과들에 대한 오차를 비교하였다.

표 5 구간 1의 과거 위치 추정

| 샘플 구간 | 샘플 간격 (km) | 오 차 | |
|-------|------------|------------|------------|
| | | 선형 함수 | 스플라인 함수 |
| 구간1 | 0.2 | 0.21306819 | 0.22515947 |
| | 0.4 | 0.50710225 | 0.47740868 |
| | 0.6 | 0.73994255 | 0.69263140 |
| | 0.8 | 1.12428980 | 1.00089700 |
| | 1.0 | 1.36852940 | 1.19549780 |
| | 1.2 | 1.63591250 | 1.56438090 |
| | 1.4 | 2.06228760 | 1.99180980 |
| | 1.6 | 3.06960230 | 2.76483060 |

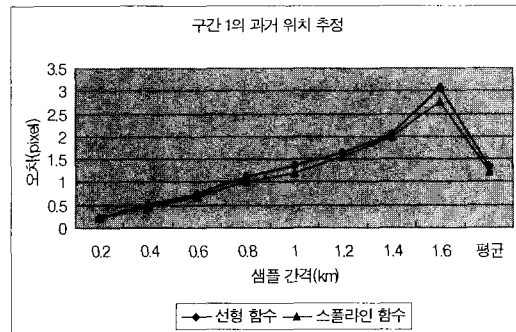


그림 10 구간 1의 오차 비교 그래프

표 5와 그림 10은 구간 1의 각 샘플 간격에 대해 선형 함수와 스플라인 함수를 이용한 과거의 위치 추정 결과에 대한 오차 값을 비교한 것이다. 오차의 계산은 추정된 위치 값과 실제 표 3, 4의 이력 정보 릴레이션에 저장된 위치 값과의 절대오차를 구하는 방법을 사용하였다. 그리고, 각각의 샘플 간격에 대해 표 3, 4의 완전한 이동 경로 릴레이션에 저장된 모든 과거 시점에 대해 위치를 추정하였다. 표 5에서 약 0.2km의 샘플 간격에서만 선형 함수에 의한 위치 추정 오차가 적게 나타난 것을 알 수 있으며, 평균적으로 스플라인 함수에 의한 위치 추정 오차가 적게 발생됨을 알 수 있었다.

표 6 구간 2의 과거 위치 추정

| 샘플 구간 | 샘플 간격 (km) | 오 차 | |
|-------|------------|-----------|------------|
| | | 선형 함수 | 스플라인 함수 |
| 구간2 | 0.2 | 0.2315341 | 0.23694874 |
| | 0.4 | 0.5937500 | 0.55593234 |
| | 0.6 | 0.8055555 | 0.78506120 |
| | 0.8 | 1.3863636 | 1.31954980 |
| | 1.0 | 1.7100000 | 1.71399520 |
| | 1.2 | 2.0262895 | 1.95836680 |
| | 1.4 | 2.4725761 | 2.32099300 |
| | 1.6 | 2.6821733 | 2.55664300 |

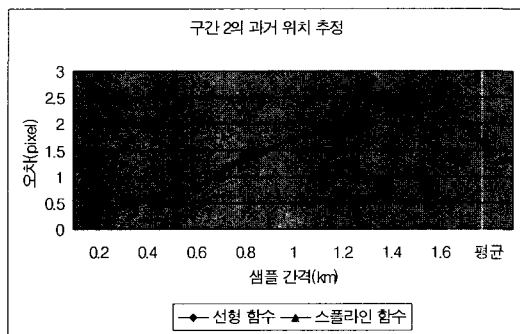


그림 11 구간 2의 오차 비교 그래프

표 6과 그림 11은 구간 2의 각 샘플 간격에 대한 과거의 위치 추정 오차의 결과이다. 구간 2에서는 약 0.2km와 1.0km의 샘플 간격에서 선형 함수에 의한 위치 추정 오차가 약간 낮게 나타났으며, 거의 모든 샘플 간격에 대해 스플라인 함수에 의한 위치 추정 오차가 더 적게 발생됨을 알 수 있었다. 구간 1과 구간 2에 대한 과거의 위치 추정 실험을 통해 선형 보간법 보다는 이 논문에서 제안한 부분구간 3차 스플라인 함수에 의한 과거의 위치 추정 오차가 더 낮게 발생됨을 알았다.

이 결과를 통해 실험에 사용된 그림 8, 9와 같이 비교적 곡선의 형태로 그 위치가 변화되는 이동 경로에서는 직선의 형태로 표현되는 선형 보간법에 의한 위치 추정보다는 부분구간에 대해 서로 다른 3차 스플라인 보간 다항식을 이용한 위치 추정 방법이 좀더 정확한 결과를 얻을 수 있음을 확인하였다. 특히, 위치 샘플의 간격이 커질수록 스플라인 함수에 의한 위치 추정 결과가 더 정확함을 알 수 있었다.

4.2 미래의 위치 추정

미래의 위치 추정 실험에 사용된 이동 경로 텔레이션은 과거의 위치 추정에 사용된 표 3과 4의 구간 1, 구

간 2의 이력 정보 릴레이션과 동일하다. 3절에서 제안한 미래의 위치 추정 방법을 이용한 추정 결과의 오차 분석을 위해, 그림 7의 미래 위치 추정 알고리즘에서 사용된 과거의 이력 객체를 나타내는 변수 N의 값을 2 ~ 10까지 변화시켜가며 위치를 추정하였다. 실험에서 사용된 이동 객체의 위치 샘플 간격은 약 100m 정도이다. 미래의 위치 추정에 사용된 임의의 미래 시점은 표 3, 4의 이력 릴레이션에 저장된 시간 값 중에서 임의로 선정하였다. 그리고, 추정된 위치와 이력 릴레이션에 저장된 위치 값의 절대오차를 계산하였으며, 각 이력 객체 수에 대해 이를 비교하였다.

표 7과 그림 12는 구간 1에 대해 이력 객체의 수를 2 ~ 10까지 변화시키며 미래 시점에 대해 위치를 추정한 결과이다. N의 값이 3일 때 추정된 결과와 표 3의 이력 정보 릴레이션에 저장된 위치와의 오차가 가장 적게 나타남을 알 수 있었다.

표 7 구간 1의 미래 위치 추정

| 샘플 구간 | 이력 객체 수 | 오 차 |
|-------|---------|------------|
| 구간1 | 2 | 0.92200470 |
| | 3 | 0.87643254 |
| | 4 | 0.91426300 |
| | 5 | 0.93682367 |
| | 6 | 0.94687310 |
| | 7 | 0.97437525 |
| | 8 | 0.98521435 |
| | 9 | 0.99914810 |
| | 10 | 1.03034500 |

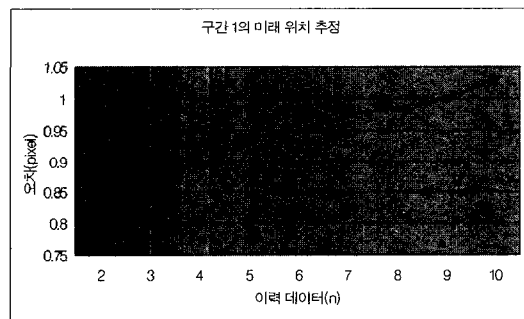


그림 12 구간 1의 오차 비교 그래프

표 8과 그림 13은 구간 2에 대한 미래 시점의 위치 추정 결과에 대한 오차 값과 그래프이다. 구간 1에서와 마찬가지로 구간 2에서도 N의 값이 3일 때 위치 추정

표 8 구간 2의 미래 위치 추정

| 샘플 구간 | 이력 객체 수 | 오 차 |
|-------|---------|------------|
| 구간1 | 2 | 0.96820810 |
| | 3 | 0.93895350 |
| | 4 | 0.96783614 |
| | 5 | 0.99779415 |
| | 6 | 1.03077030 |
| | 7 | 1.06894870 |
| | 8 | 1.08426100 |
| | 9 | 1.10015060 |
| | 10 | 1.11515160 |

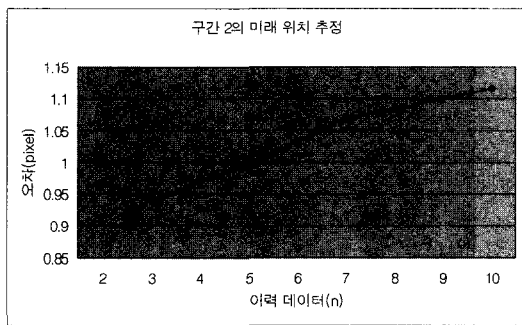


그림 13 구간 2의 오차 비교 그래프

오차가 가장 적게 나타남을 알 수 있었다. 이력 정보의 수가 많을수록 위치 추정 결과가 더 정확하게 발생될 수도 있지만, 이 실험에서 사용된 구간 1, 2와 같은 이동경로에서는 N의 값이 4가 넘으면 점점 오차가 더 크게 발생되었다. 특히, N이 2가 되면 제안한 위치 예측 함수는 선형 보간법과 동일한 식이 된다. 따라서 실험 결과를 통해 N이 3일 경우에는 선형 보간법보다 제안한 위치 예측 함수가 더 적은 오차를 발생시킴을 알 수 있었다. 그러나 모든 이동경로에 대해 이력 객체의 수가 3일 때 오차가 가장 적게 나타남을 의미하지는 않는다. 이동 객체의 응용 분야에 적용되는 이동경로의 특성에 따라 N의 값은 달라지는 특징을 가진다.

5. 결론

이동 객체의 불확실성은 데이터베이스의 모델링, 질의 처리, 인덱싱, 질의 결과의 부정확성 등과 같은 여러 가지 문제들을 발생시킨다. 데이터의 불확실성으로 인해 발생하는 질의 결과의 부정확성은 사용자에게 잘못된 의사 결정 요인을 제공할 수도 있다. 지금까지의 불확실성 처리에 대한 연구들은 과거 또는 미래의 불확실한 위치 추정 중 한 가지에만 주로 초점을 맞추어 연구하

였기 때문에, 동일한 데이터를 가지고 동시에 과거와 미래의 이동 위치를 추정하는 방법을 제시하지 못하였다.

따라서 이 논문에서는 동일한 데이터베이스에 저장된 과거의 이동 정보를 이용하여 이동 객체의 과거 및 미래의 불확실한 위치를 추정하는 방법과 그 알고리즘을 제안하였다. 데이터베이스에 저장되지 않은 과거의 임의의 시점에 대한 위치 추정을 위해 3차 스플라인 보간법을 사용하였으며, 미래의 임의의 시점에 대한 위치 추정에는 과거 이동 정보의 평균값을 이용하였다.

제안 방법의 특성 분석을 위해 이동 경로와 위치 샘플을 발생시켜 이력 릴레이션에 저장한 후, 과거 및 미래의 위치 추정 실험을 하였다. 이 실험을 통해 제안한 부분구간 3차 스플라인 함수를 이용한 과거의 위치 추정 결과가 거의 모든 샘플 간격에서 선형 보간법을 이용한 결과보다 오차가 적게 발생됨을 알 수 있었다. 또한 과거 이력 정보의 수를 변화시켜가며 미래의 위치 추정을 실험한 결과, 이력 정보의 수가 3일 때 위치 추정 오차가 가장 적게 나타남을 알 수 있었다. 특히, N의 값이 2인 경우는 선형 보간법과 동일한 함수가 되므로, N이 3일 때는 제안한 위치 예측 함수가 선형 보간법보다 더 적은 오차를 발생시킴을 알았다.

이 논문에서는 데이터베이스에 저장된 이력정보를 이용하여 이동 객체의 과거 및 미래의 위치 추정 실험을 하였으며, 위치 추정 실험 결과의 정확성에 초점을 맞추어 분석하였다. 대부분의 이동 객체 관리 시스템은 실시간 환경을 고려한 응용 서비스에 적용되고 있다. 현재 이 논문에서 제안한 방법을 실시간 환경의 e-logistics 응용 시스템에 적합한 형태로 확장하는 연구가 진행중이다. 앞으로는 이 논문에서 제안한 알고리즘이 실시간 환경에서 빠른 처리 속도를 지원할 수 있도록 하는 방법이 연구될 것이다. 특히 미래의 위치 예측 함수는 위치 샘플의 간격 및 이동 경로의 특성에 따라 적용 방법이 달라질 수 있으므로 향후 더 많은 연구가 필요한 분야이다.

참 고 문 헌

[1] A.P.Sistla, O.Wolfson, S.Chamberlain, and S.Dao, "Querying the Uncertain Position of Moving Objects," Springer Verlag Lecture Notes in Computer Science number 1399, 1998, pp.310-337.
 [2] O.Wolfson, S.Chamberlain, S.Dao, L.Jiang, and G.Mendez, "Cost and Imprecision in Modeling the Position of Moving Objects," Proceedings of the 14th International Conference on Data Engineering, Orlando, FL, Feb. 1998.

- [3] O.Wolfson, B.Xu, S.Chamberlain, and L.Jiang, "Moving Objects Databases: Issues and Solutions," Proceedings of the 10th International Conference on Scientific and Statistical Database Management, SSDBM'98, Capri, Italy, Jul. 1998. pp. 111-122.
- [4] D.Pfoser and C.S.Jensen, "Capturing the Uncertainty of Moving Object Representations," CHOROCHRONOS, Technical Report CC-99-2, Apr. 1999.
- [5] D.Pfoser and N.Tryfona, "Fuzziness and Uncertainty in Spatiotemporal Applications," CHOROCHRONOS, Technical Report CH-00-04, Feb. 2000.
- [6] A.P.Sistla, O.Wolfson, S.Chamberlain and S.Dao, "Modeling and Querying Moving Objects," Proceedings of the International Conference on Data Engineering, 1997, pp. 422-432.
- [7] M.Erwig, R.H.Guting, M.Schneider and M.Vazirgiannis, "Spatio-Temporal Data Types : An Approach to Modeling and Querying Moving Object in Databases," GeoInformatica Vol. 3, No. 3, 1999, pp. 269-296.
- [8] R.H.Guting, M.H.Bohlen, M.Erwig, C.S.Jensen, N.A.Lorentzos, M.Schneider, and M.Varzirgiannis, "A Foundation for Representing and Querying Moving Objects," ACM Transactions on Database Systems, Vol. 25, No. 1, Mar. 2000, pp. 1-42.
- [9] J. Moreira, C. Ribeiro, and J.M. Saglio, "Representation and Manipulation of Moving Points: An Extended Data Model for Location Estimation", Cartography and Geographic Information Systems(CaGIS), ACSM, Vol. 26, No. 2, Apr. 1999.
- [10] I.B.Oh, Y.A.Ahn, E.J.Lee, K.H.Ryu, and H.G.Kim "Prediction of Uncertain Moving Object Location," Proceedings of International Conference on East-Asian Language Processing and Internet Information Technology, EALPIIT 2002, Jan. 2002, pp. 51-58.
- [11] M.H.Huh, Y.A.Ahn, and K.H.Ryu, "Moving Object Location Change Function using Cubic Spline Interpolation," Proceedings of the 6th World Multiconference on Systemics, Cybernetics and Informatics, SCI 2002, Vol. V, Computer Science I, Jul. 2002, pp. 336-341.
- [12] L.Forlizzi, R.H.Guting, E.Nardelli, and M.Schneider, "A Data Model and Data Structures for Moving Objects Databases," Proceedings of the ACM SIGMOD'00, pp. 319-330.
- [13] S.Shumilov and J.Seibeck, "Database Support for Temporal 3D Data : Extending the GeoToolKit," Proceedings of the 7th EC-GI & GIS Workshop, ECGIS'01, Potsdam, Germany, Jun. 2001.
- [14] 정상권, 이승래, 권육현, "CEMTool을 활용한 수치해석," <http://icat.snu.ac.kr:3000>, 2001.
- [15] 김하진, 김민구, 윤경현, "구간적 3차 보간 다항식에 의한 곡면 구성에 관하여," 한국정보과학회 논문지, Vol. 10, No. 3, 1983년 8월, pp. 182-189.
- [16] R.Bartels, J.Beatty, and B.Barsky, "An Introduction to Splines for Use in Computer Graphics & Geometric Modeling," Morgan Kaufmann Publishers, Inc., 1987.
- [17] O.Wolfson, P.Sistla, B.Xu, J.Zhou, S.Chamberlain, N.Rishe, and Y.Yesha, "Tracking Moving Objects Using Database Technology in DOMINO," Proceedings of NGITS'99, The 4th Workshop on Next Generation Information Technologies and Systems, Zikhron-Yaakov, Israel, Jul. 1999, pp. 112-119.
- [18] S.S.Park, Y.A.Ahn, and K.H.Ryu, "Moving Objects Spatiotemporal Reasoning Model for Battlefield Analysis," Proceedings of Military, Government and Aerospace Simulation part of ASTC'01, Apr. 2001, pp. 108-113.

안 윤 애



1993년 한남대학교 전자계산공학과(공학사). 1996년 충북대학교 대학원 전자계산학과(이학석사). 1999년 ~ 현재 충북대학교 대학원 전자계산학과 박사과정. 관심분야는 이동 객체 데이터베이스, 시공간 데이터베이스, 지리정보 시스템, 지식

기반 정보검색 시스템

류 근 호

정보과학회논문지 : 데이터베이스
제 29 권 5호 참조