

스트리밍 환경에서 다항식 기반의 이동객체 위치 추정

이원철, 문양세, 이상민, 노희영
강원대학교 컴퓨터 과학과
{woncheol⁰, ysmoon, smrhee, rohhy}@kangwon.ac.kr

Polynomial-based Estimation of Moving Object Trajectories in Stream Environment

Won-Cheol Lee⁰, Yang-Sae Moon, Sang-Min Rhee, and Hi-Young Roh
Dept. of Computer Science, Kangwon National University.

요 약

본 논문은 스트림 형태로 발생하는 이동객체의 위치정보에 대해 과거 시점의 위치정보 추정 방법을 제안한다. 기존 이동객체의 위치 추정에 대한 연구는 메모리 량의 제한이 없는 상태에서, 이미 저장된 과거 데이터를 이용하여 임의의 과거 시점 위치를 추정하는 방법이다. 그러나 스트리밍 환경에서는 위치정보가 무한하게 발생하기 때문에 모든 위치정보를 저장 및 관리할 수 없다. 따라서 본 논문에서는 스트림 형태로 발생하는 위치정보에 대하여 제한된 메모리를 사용하여 임의의 과거시점 위치를 추정하는 방법을 제안한다. 이를 위하여, 실제 위치, 무제약 추정위치, 제약 추정위치의 세가지 위치 개념을 정형적으로 정의하고, 다항식을 이용하여 이들 위치를 추정하는 체계적인 방법을 제안한다.

1. 서 론

시간의 흐름에 따라 연속적인 위치 변화를 발생하는 객체를 이동객체라 한다. 이러한 이동객체는 주로 동물군 관찰 시스템, 기상 관측, 교통통제 시스템, 비디오 데이터 계열 검색 분야에서 연구 되었다. 최근에는 위치 기반 서비스 및 텔레메틱스 기술을 이용한 연구가 활발히 진행되고 있다[1,3]. 이 이동객체의 위치 정보는 스트림 데이터와 같이 끊임없이 위치정보를 발생한다[4].

이동객체의 위치정보가 스트림 형태로 발생하는 환경에서는 제한된 저장공간을 사용하여 이동객체의 위치를 관리해야 한다. 따라서, 스트리밍 환경기반의 과거 위치 추정방법은 기존의 연구방법과 차이가 있다. 즉, 기존의 연구방법은 이동 객체의 위치정보를 제한없이 저장하고, 위치를 검색하고 예측하였다. 반면에 스트리밍 환경에서는 저장 및 관리할 수 있는 메모리량에 제한이 있으며, 이에 따라 저장해야 하는 위치정보와 추정방법이 기존 연구와는 달라야 한다.

본 논문에서는 이와 같이 스트림 형태로 발생하는 위치정보를 제한된 메모리에 저장하고, 과거 위치를 추정하는 방법을 제안한다. 즉, 제한된 메모리를 초과하여 저장되지 않도록 하기 위하여 기존의 위치정보에서 새로운 위치정보를 추출하여 저장한다. 또한 저장된 데이터를 기반으로 과거의 위치를 추정하는 방법을 제안한다. 본 논문에서는 새로운 위치정보 추출 및 과거 위치 추정을 위해 다항식 기반의 다양한 예측방법을 제안한다.

본 논문의 구성은 다음과 같다. 제2장에서는 관련연구를 설명한다. 제3장에서는 본 논문에서 다루는 문제를 정의한다. 제4장에서는 다항식을 이용한 추정 및 저장방법을 제안한다. 마지막으로, 제5장에서 결론과 향후 연구를 기술한다.

2. 관련연구

본 논문에서는 스트림 형태로 발생하는 이동객체의 위치 정보를 추정하는 방법을 다룬다. 이와 관련된 연구로는 이동객체의 저장과 관리방법, 궤적 추정 방법, 시계열 데이터베이스의 유사 시퀀스 매칭 연구 등이 있다.

전통적인 이동 객체 연구에서는 주로 객체의 현재 위치, 과거 위치(궤적), 미래 위치를 저장하거나 추정하는 내용을 다루었다[3,6]. 특히, 객체의 위치를 빠르게 검색하기 위한 색인 기법이 주로 연구되었으며, 이들 색인 방법은 다시 현재 위치를 검색하기 위한 방법과 미래 위치를 예측하기 위한 방법으로 분류할 수 있다.

이동 객체의 위치를 추정하는 방법으로는 과거 시점의 궤적을 추정하는 방법과 미래 시점의 위치를 예측하는 방법이 연구되었다. 과거 궤적을 추정하는 방법은 이동 객체의 저장된 두 개 이상의 과거 위치를 사용하여, 저장되지 않은 시점의 위치(궤적)를 추정하는 방법이다. 이러한 과거 궤적 추정 방법은 다시 직선(line) 기반의 추정방법과 곡선(curve) 기반의 추정 방법으로 구분할 수 있다. 직선 기반의 궤적 추정방법은 이동

객체가 갖는 임의의 움직임에 따른 복잡성을 피할 수 있고, 움직임의 형태를 단순하게 추정할 수 있는 장점이 있다[6]. 그러나 실제 세계에서 이동 객체는 직선이라고 보다는 복잡하고 다양한 이동 패턴을 가지기 때문에, 직선 기반의 추정 방법은 실제의 문제를 제대로 해결할 수 없다. 이를 해결하기 위하여 제안된 방법이 곡선 기반의 궤적 추정 방법이다.

시계열 데이터베이스의 유사 시퀀스 매칭에서 연구된 저차원 변환(lower-dimensional transformation)도 같은 위치 변경이 일어나는 이동객체의 과거 혹은 미래 위치 추정에 활용될 수 있다. 즉 이동객체의 위치를 시계열 데이터 형태로 나타내고, 이를 저차원 변환하여 저장, 관리, 예측에 활용하는 것이다.

3. 문제 정의

이동객체의 위치 정보가 스트림 환경에서는 지속적으로 업데이트 되고, 특정 시점에서의 위치는 다차원 데이터로 표현할 수 있다. 그러나 다차원 데이터는 계산이 너무 복잡하고 추론하기 매우 어렵다. 그러므로 시간의 흐름에 따라 지속적으로 변화하는 이동객체의 위치를 특정 차원을 중심으로 표현하면, 각 시간 값과 각 축 값을 쌍으로 표현할 수 있다. 이를 설명하기 위해 이동객체를 O 라 하고, O 의 2차원 위치 정보를 $O(x, y)$ 로 표현한다. 그리고 객체 O 는 시간의 흐름에 따라 이동하므로, 객체 O 를 $O(t, x, y)$ 로 표현할 수 있다. 또한 이동객체 O 의 x 축 및 y 축 위치 정보를 시간 축으로 구분하여, x 축의 경우 $O(t, x)$, y 축의 경우 $O(t, y)$ 로 나타낼 수 있다. 표 1은 이러한 표기법에 따른 객체 O 의 위치 변화를 나타낸 것이다.

표 1. 객체 O 의 위치 데이터 예제

구분	위치 데이터
x 축 및 y 축	(1, 2.1, 2.2), (2, 1.0, 2.5), (3, 2.2, 4.0), (4, 4.1, 5.3), (5, 5.3, 3.4) ……
x 축	(1, 2.1), (2, 1.0), (3, 2.2), (4, 4.1), (5, 5.3) ……
y 축	(1, 2.2), (2, 2.5), (3, 4.0), (4, 5.3), (5, 3.4) ……

이와 같이 모든 다차원 위치 데이터는 시간 축을 기준으로 하는 이차원 데이터로 표현할 수 있고, 따라서 본 논문에서는 이동객체의 위치가 (시간 값, 위치 값)의 쌍, 특히 (t, x) 와 같이 2차원으로 나타낸다고 가정한다.

이동객체의 위치가 보고되는 시점을 $t_i (1 \leq i \leq n)$ 라 하면, 이동객체의 위치 정보는 시점 t_i 에 지속적으로 업데이트 된다. 본 논문에서는 이동객체 위치 정보를 객체가 실제 이동했던 위치인지, 아니면 추정된

위치인지의 여부에 따라 실제위치 또는 추정위치로 구분한다. 또한, 추정위치는 위치 정보를 저장할 수 있는 메모리 량의 제한 여부에 따라 무제약추정위치와 제약추정위치로 구분한다. 이들을 다음과 같이 정형적으로 정의한다.

정의 1: 이동객체 O 에 대해 보고된 위치 데이터가 $\{(t_1, x_1), (t_2, x_2), \dots, (t_n, x_n)\}$ 라 할 때, 임의의 시점 $t_k (\neq t_i, 1 \leq k \leq n)$ 에서의 실제위치(actual position)란, 보고 및 저장되지는 않았으나 이동객체 O 가 시점 t_k 에 존재했던 실제 위치를 의미한다. □

정의 2: 이동객체 O 에 대해 보고된 위치 데이터가 $\{(t_1, x_1), (t_2, x_2), \dots, (t_n, x_n)\}$ 라 할 때, 임의의 시점 $t_k (\neq t_i, 1 \leq k \leq n)$ 에서의 무제약추정위치(unconstraint estimated position)란, n 개의 모든 (t_i, x_i) 위치 데이터를 저장한 상태에서 추정한 시점 t_k 에서의 이동객체 O 의 위치이다. □

정의 3: 이동객체 O 에 대해 보고된 위치 데이터가 $\{(t_1, x_1), (t_2, x_2), \dots, (t_n, x_n)\}$ 라 할 때, 임의의 시점 $t_k (\neq t_i, 1 \leq k \leq n)$ 에서의 제약추정위치란(constraint estimated position), 오직 m 개 ($m \ll n$)의 위치 데이터 $(t_j, x_j) (j = i \vee j \neq i)$ 를 저장한 상태에서 추정한 시점 t_k 에서의 이동객체 O 의 위치이다. □

스트림 환경에서는 이동객체의 위치를 저장할 수 있는 메모리 량에 한계가 있다. 따라서, 본 논문에서는 메모리 량이 제한된 스트림 환경에서 정의 3의 제약추정위치를 얻는 문제를 다룬다.

정의 4: 스트림 환경에서 이동객체의 과거위치추정이라, 하나의 객체에 대해 위치정보를 저장할 수 있는 메모리 량이 제한되어 있을 때, 지속적으로 변화하는 객체의 위치 데이터를 반영하면서 임의의 시점에서의 제약추정위치를 계산하는 방법이다. □

본 논문에서는 다항식을 사용하여 정의 4의 스트림 환경에서 이동객체의 제약추정위치를 계산하는 방법을 제안한다.

4. 다항식 기반 위치 추정 방법

본 장에서는 스트림 환경에서 이동객체의 제약추정위치를 계산하기 위하여, 다항식을 이용하는 전체적인 프레임워크를 제안한다.

4.1 제안 프레임워크

프레임워크는 크게 두 가지 과정으로 구성된다. 첫째는 제한된 메모리 량으로, 지속적으로 변화하는 객체의

위치 데이터를 저장하는 과정이다. 둘째는 제한된 위치 데이터를 사용하여 과거 임의 시점의 제약추정위치를 계산하는 과정이다

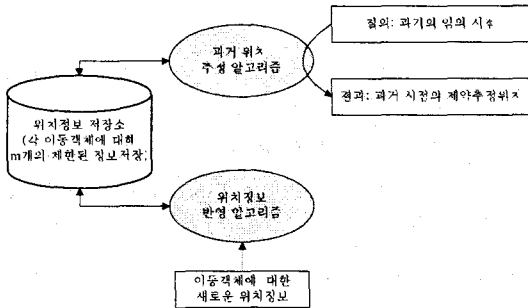


그림 1. 제한된 프레임워크

그림 1은 스트림 환경에서 제약추정위치를 계산하는 전체 프레임워크를 나타낸다. 그림을 보면, 우선 위치정보 저장소에는 각 이동객체에 대해 m 개의 제한된 정보를 저장관리한다. 그리고, 위치정보 반영 알고리즘은 이동객체의 위치가 변경되었을 때, 해당 객체에 대해 관리하는 m 개의 정보를 새롭게 변경하는 기능을 수행한다. 즉, 객체가 이동하여 새로운 위치가 보고되었을 때, 이를 저장된 m 개의 정보에 반영하기 위한 알고리즘이다. 다음으로, 과거 위치추정 알고리즘은 사용자가 제시한 과거 임의 시점에 대한 제약추정위치를 계산한다. 즉, 위치 정보 저장소에 저장된 m 개의 정보를 사용하여 사용자가 제시한 시점의 제약추정위치를 계산하여 반환하는 것이다.

4.2 추정위치 계산 방법

이동객체 O 에 대해 보고된(혹은 저장된) 위치 데이터가 $\{(t_1, x_1), (t_2, x_2), \dots, (t_m, x_m)\}$ 이라 할 때, 추정 위치를 계산하는 방법은 임의의 시점 t_k 에 대해 객체 O 의 위치 x_k 를 계산하는 것이다. 이때, 객체의 모든 위치 정보를 저장할 수 있다면 x_k 는 무제약추정위치가 되고, 제한된 위치 데이터만 저장할 수 있다면 x_k 는 제약추정 위치가 된다. 본 논문에서는 이러한 (무)제약추정위치를 계산 방법으로 다항식[1-3,6]을 사용한다. 특히 주어진 시점과 인접한 두 개의 위치를 사용하는 직선기반 추정 방법과 세 개 이상의 위치를 사용하는 곡선기반 추정 방법을 제안한다.

직선기반 추정방법

직선기반 추정 방법[3,6]은 임의의 시점 t_k 에 대해 인접한 두 개의 저장된 시점 t_i 와 t_{i+1} ($t_i \leq t_k \leq t_{i+1}$) 의 위치정보 x_i 와 x_{i+1} 을 사용하는 방법이다. 즉, 두

점 (t_i, x_i) 와 (t_{i+1}, x_{i+1}) 을 2차원 좌표상에서 직선으로 연결한 방정식 $f(t)$ 을 계산한다. 그런 다음 방정식 $f(t)$ 에 질의 시점 t_k 을 대입하여 추정위치 x_k 를 $f(t_k)$ 로 추정하는 것이다. 2차원 좌표의 두 지점 (t_i, x_i) 와 (t_{i+1}, x_{i+1}) 이 주어졌을 때, 이 두 좌표 값을 지나는 직선의 방정식은 다음 공식 (1)과 같이 구할 수 있다.

$$f(t) = \frac{(t_{i+1} - t) \cdot x_i + (t - t_i) \cdot x_{i+1}}{t_{i+1} - t_i} \quad (1)$$

곡선기반 추정방법

곡선 기반 추정 방법[1,2]은 두개의 인접한 위치 데이터 대신, 세 개 이상의 위치 정보를 사용한다. 문제를 간단히 하기 위하여, 본 논문에서는 주어진 m 개의 위치 데이터를 모두 이용한다. 즉, 저장된 m 개의 위치 정보 $\{(t_1, x_1), (t_2, x_2), \dots, (t_m, x_m)\}$ 을 지나는 $(m-1)$ 차 다항식 $f(t)$ 을 구하고, 이 다항식에 주어진 시간 값을 대입하여 위치 정보를 추정하는 방법이다. 저장된 좌표 값들은 등간격 ($t_{i+1} - t_i = t_i - t_{i-1}$) 이라는 보장이 없기 때문에 Chebyshev, Netwon 다항식 등은 사용할 수 없다. 따라서, 본 논문에서는 등간격이 아닌 경우에도 다항식 계산이 가능한 라그랑제 기초 보간함수(Lagrange interpolation basis function)[5]을 사용한다. 좌표상에서 m 개의 위치정보 $\{(t_1, x_1), (t_2, x_2), \dots, (t_m, x_m)\}$ 이 주어졌을 때, 이 좌표 값들을 지나는 곡선의 라그랑제 방정식은 다음 공식 (2)와 같이 구할 수 있다.

$$f(t) = x_0 L_0(t) + x_1 L_1(t) + \dots + x_m L_m(t) \quad (2)$$

$$L_i(t) = \frac{(t-t_1)(t-t_2) \dots (t-t_m)}{(t_i-t_1)(t_i-t_2) \dots (t_i-t_m)}$$

4.3 위치정보 추출방법

본 절에서는 메모리에 제약이 따르는 스트림 환경에서, 제한된 m 개의 정보를 어떻게 추출하는지의 방법을 제안한다. 즉, 위치정보 추출이란 m 개의 제한된 메모리에 저장된 위치정보 $\{(t_1, x_1), (t_2, x_2), \dots, (t_m, x_m)\}$ 에 새로운 위치정보 (t_{m+1}, x_{m+1}) 이 입력되었을 때, 위치정보 (t_{m+1}, x_{m+1}) 을 반영한 m 개의 새로운 위치정보 $\{(t'_1, x'_1), (t'_2, x'_2), \dots, (t'_m, x'_m)\}$ 을 계산하는 방법이다. 위치정보 추출방법으로 본 논문에서는 시간 축을 기준으로 균등간격으로 위치정보를 추출하는 방법과 균등하지 않은 간격으로 위치정보를 추출하는 방법을

사용한다. 또한 균등하지 않은 경우는 다시 기울기 기반 위치정보 추출방법과 최근 시점 강조 위치정보 추출방법으로 나누어 제한한다.

균등간격 정보추출 방법

균등간격이란 m 개의 새로운 위치정보를 추출할 때, 시간 축을 기준으로 동일한 시간 간격으로 위치정보를 추출하는 방법이다. 즉, 위치정보 추출 시점 t'_i 와 시점 t'_{i-1} 의 간격이 시점 t'_{i+1} 과 t'_i 의 간격과 동일한 것이다. 균등간격 추출에 대해 직선기반 방법과 곡선기반 방법을 구분하여 설명한다.

직선기반 방법은 현재까지의 $m+1$ 개 위치정보에서 새로운 시간간격 $(t'_{i+1}-t'_i)$ 을 계산하고, 이를 공식 (1)에 적용하여 새로운 m 개의 위치정보를 계산하는 방법이다. 이를 위해, 새로운 위치 정보 (t_{m+1}, x_{m+1}) 가 입력되었을 때, 마지막 입력시간 t_{m+1} 과 처음 입력시간 t_1 의 차이를 간격의 수 $m-1$ 로 나누어 시간간격 δ 를 계산한다. 그리고 시간간격 δ 를 기준으로 새로운 시점 $t'_i (=t_1 + (i-1) \cdot \delta)$ 를 계산한다. 다음으로, 각 시점 t'_i 의 위치 x'_i 를 계산하기 위하여, t'_i 과 인접한 시간 쌍 t_j 와 t_{j+1} ($t_j \leq t'_i \leq t_{j+1}$)을 찾아, (t_j, x_j) 와 (t_{j+1}, x_{j+1}) 을 잇는 직선의 방정식 $f(t)$ 를 공식 (1)로 계산한다. 그런 다음, 직선의 방정식 $f(t)$ 에 시점 t'_i 를 대입하여 x'_i 를 $f(t'_i)$ 로 계산한다. 그러나 x'_1 과 x'_m 은 예외처리하여 계산하지 않는다. 이는 처음 입력시점 t_1 이 새로운 시점 t'_1 과 동일하고, 마지막 입력시점 t_{m+1} 도 새로운 시점 t'_m 과 동일하기 때문이다. 따라서 새로운 시점 t'_1 과 t'_m 에 대한 x 좌표 값 x'_1 과 x'_m 은 각각 x_1 과 x_{m+1} 로 설정한다.

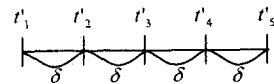
곡선기반 방법은 직선 방정식 대신 라그랑제 다항식을 사용하는 것을 제외하고는 직선기반 방법과 동일하다. 곡선기반 방법의 절차는 다음과 같다. 새로운 위치 정보 (t_{m+1}, x_{m+1}) 가 입력 되었을 때, $m+1$ 개의 위치정보를 사용한 라그랑제 다항식 $f(t)$ 를 구한다. 그런 후 직선기반 방법과 동일한 방식으로 시간 간격 δ 를 계산한다. 그리고 δ 를 사용하여 새로운 시점 $t'_i (=t_1 + (i-1) \cdot \delta)$ 를 계산한다. 다음으로, 각 시점 t'_i 의 위치 x'_i 를 계산하기 위하여 앞서 계산된 라그랑제 다항식을 사용한다. 즉, 새로운 시점 t'_i 를 라그랑제 다항식 $f(t)$ 에 대입하여 새로운 위치정보 x'_i 를 계산한다. 단, x'_1 과 x'_m 은 직선 기반 방법과 같이 예외처리하여 계산하지 않는다.

기울기 기반 데이터 추출 방법

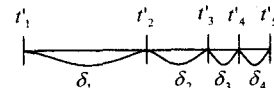
기울기 기반 추출 방법은 비균등 간격 위치정보 추출 방법으로서, 기울기 변화가 심한 구간에서는 많은 수의 위치정보를 추출하고 그렇지 않은 구간에서는 적은 수의 위치정보를 추출한다. 즉, 기울기 변화가 심하지 않은 구간은 적은 수의 정보로 간략히 표현하고, 변화가 심한 구간에서는 많은 수의 정보로 자세히 표현하는 것이다. 이 방법은 직선기반 방법에 적용이 가능하며, 그 절차는 다음과 같다. 새로운 위치정보 (t_{m+1}, x_{m+1}) 가 입력되었을 때, 각각의 인접한 위치정보 (t_i, x_i) 와 (t_{i+1}, x_{i+1}) 에 대한 기울기 값 $\alpha_i = (x_{i+1} - x_i)/(t_{i+1} - t_i)$ 을 계산한다. 그런 다음, 인접한 두 기울기 α_i 와 α_{i+1} 의 차이 $|\alpha_i - \alpha_{i+1}|$ 을 계산하여, 그 차가 가장 작은 구간을 찾아 계산하는데 포함된 (t_{i+1}, x_{i+1}) 을 삭제한다. 단, 기울기 차이가 가장 작은 구간이 여러 개 발생할 경우, 그 구간들 중에서 가장 과거 구간에 포함된 위치정보를 삭제한다. 그 결과로 남은 나머지 m 개의 위치정보는 다시 저장한다

최근 시점 강조 위치정보 추출 방법

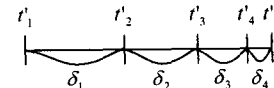
이 방법은 이동객체 데이터베이스 응용에서, 최신의 위치정보는 자주 이용하지만, 과거의 위치정보는 거의 이용하지 않는 특성을 고려한 방법이다. 즉, 최신의 위치정보는 많은 수를 추출하고, 과거의 위치정보는 적은 수를 추출하는 방법이다. 따라서 최근 시점 강조 데이터 추출 방법은 비균등 위치정보 추출 방법으로, 과거 시점에 대해 다른 비율로 위치정보를 추출한다.



(a) 균등간격 정보 추출



(b) 비균등간격 정보추출



(c) 비균등간격 정보추출

그림 2. 간격설정에 따른 정보 추출 시점의 변화

비균등 간격 데이터 추출을 위해서는 위치 정보를 추출하는 시점을 달리해야 한다. 그림 3은 균등 간격과 비 균등 간격 정보 추출에 따른 시점 변화를 나타낸다. 그림 3(a)는 균등 간격으로 각 시점의 시간 간격이 δ 로 동일하다. 반면에, 그림 3(b)는 비균등 간격으로 각 시점의 시간 간격 δ_i 가 동일하지 않다. 따라서, 비균등 간격 정보추출을 위해서는 각 시점간의 간격인 δ_i 를 구해야 한다. 시간 간격 δ_i 가 일정한 규칙을 가져야 정형적인 모델링이 가능하므로, δ_i 가 $f(t)$ 와 같이 함수로 나타낸다고 가정한다. 즉, $\delta_i = f(t)$ 라 했을 때, 다음식 (3)에 의해 새로운 시점 t'_i 를 구할 수 있다. 즉, $\delta_i = f(t)$ 라 했을 때, 식 (3)을 사용하여 비균등 간격 정보 추출을 위해서는 함수 $f(t)$ 를 결정해야 한다.

$$\begin{cases} t'_1 = t_1 \\ t'_i = t_1 + \sum_{i=1}^i \delta_i \quad (1 < i < m) \\ t'_m = t_{m+1} \end{cases} \quad (3)$$

함수 $f(t)$ 로는 그림 3(b)와 같이 $f(t) = (t_{m+1} - t_1) / 2^{t-1}$ 로 하거나, $f(t) = (m-i)\theta$ ($\theta = ((m-1)(m-2))/2$)로 하거나 지수 함수 등의 다른 함수를 사용하여 나타낼 수 있다. 또한 그림 3(a)의 균등 간격 추출방법은 $f(t) = (t_{m+1} - t_1) / (m-1)$ 로한 방법으로 해석할 수 있다.

앞서 제시한 비균등 간격 정보추출 방법은 균등간격과 마찬가지로 직선기반 및 곡선기반 방법에 모두 적용될 수 있다. 즉, 비균등간격으로 구한 시간 t'_i 를 직선 기반의 경우 식 (1)에 적용하고, 곡선 기반의 경우 식 (2)에 적용하면 된다.

5. 결론

본 논문에서는 스트림 형태로 발생하는 이동객체의 위치에 대해 과거 시점의 위치 추정 방법을 제안하였다. 기존의 연구는 메모리가 제한된 상황에서 과거 위치를 추정하는 연구가 이루어져 있지 않다. 따라서 본 논문에서는 스트림 환경을 고려하여 메모리량이 제한된 환경에서 과거 위치를 추정하는 문제를 정의 하였다. 그리고 과거 시점의 위치를 추정하기 위한 저장방법 및 추정방법을 제안하였다. 특히 과거 위치 추정을 위해 다항식 기반의 다양한 위치 예측 모델을 제안하였다. 향후 연구로는 실험을 통하여 제시한 방법의 우수성을 입증하는 것이다

참고 문헌

[1] Byunggu, Y., Seon, K., Thomas, B., and Ruben, G., "Curve-Based Representation of Moving Object

Trajectories," In *Proc. Int'l Database Engineering and Applications Symposium IEEE*, pp.419-425, 2004.

[2] Yunhan, C., and Raymond, N., "Indexing Spatio-Temporal Trajectories with Chebyshev Polynomials," In *Proc. Int'l Conf. on Data Management ACM SIGMOD*, Paris, France, pp.599-610, June, 2004.

[3] Saltenis, S., S. C., Leutenegger, S. T., and Lopez, M. A., "Indexing the Positions of Continuously Moving Objects," In *Proc. Int'l. Conf. on Data Management ACM SIGMOD*, Dallas, Texas, pp.331-342, June 2000.

[4] Palpanas, T., Vlachos, M., Keogh, E., Gunopulos D., and Truppel, W., "Online Amnesic Approximation of Streaming Time Series" In *Proc. Int'l Conf. on Data Engineering, IEEE*, Boston, MA, pp.338-349, Mar. 2004.

[5] Tao, Y., Faloutsos, C., Papadias, D., and Liu, B., "Prediction and Indexing of Moving Objects with Unknown Motion Patterns", In *Proc. Int'l Conf. on Data Management ACM SIGMOD*, Paris, France, pp.611-622, June, 2004.

[6] Pfoser, D., Jensen, C., and Theodoridis, Y., "Novel Approaches in Query Processing for moving Object Trajectories," In *Proc. Int'l conf. on Very Large Data Bases(VLDB)*, Cairo, Egypt, pp.395-406, Sept. 2000.

[7] Faloutsos, C., Ranganathan, M., and Manolopoulos, Y., "Fast Subsequence Matching in Time-Series Database," In *Proc. Int'l Conf. on Data Management ACM SIGMOD*, MI, USA, pp.419-429, May 1994.

[8] Chan, K., and Fu, W., "Efficient Times Series Matching by Wavelets," In *Int'l Conf. on Data Engineering*, Sydney, Australia, pp.126-133, mar. 1999.

[9] Chakrabarti, K., Keogh, J.E., Mehrotra, S., and Pazzani, J. M., "Locally Adaptive Dimensionality Reduction for Indexing Large Time Series Databases," *ACM Transactions on Database Systems*, Vol.27, No.2, pp.188-228, 2002.

[10] Keogh, J. E., and Pazzani, J. M., "An enhanced Representation of Time Series Which Allows Fast and Accurate Classification, Clustering and Relevance Feedback", In *Proc. Int'l Conf. on Knowledge Discovery and Data Mining*, New York, USA, pp.239-243, Aug. 1998.