

차량 위치 추적을 위한 이동 객체 관리 시스템의 설계

안 윤 애[†] · 김 동 호^{††} · 류 근 호^{†††}

요 약

이동 객체 관리 시스템은 사람, 동물, 자동차 등과 같이 시간에 따라 위치를 변경하는 시공간 데이터를 관리한다. 이러한 이동 객체 관리 시스템은 차량 위치 추적, 디지털 전장, 위치 기반 서비스 등에 적용된다. 그러나 기존의 이동 객체 관리 시스템은 이동 객체의 과거 및 미래의 위치정보를 개별적으로 관리하며, 불확실한 과거 및 미래의 위치 추정 방법을 구체적으로 제시하지 못한다. 따라서 이 논문에서는 이동 객체의 이력 정보를 관리할 뿐만 아니라, 데이터베이스에 저장된 이력정보를 이용한 이동 객체의 과거 및 미래의 위치 추정이 가능한 시스템을 제안한다. 이를 위해 차량 위치 추적을 위한 이동 객체를 모델링하고 이동 객체 데이터베이스 구조를 제시한다. 아울러 제안 시스템의 실행 모델을 제시하고, 이를 차량 추적을 위한 가상 시나리오에 적용한다.

Design of A Moving Object Management System for Tracking Vehicle Location

Yoon Ae Ahn[†] · Dong Ho Kim^{††} · Keun Ho Ryu^{†††}

ABSTRACT

Moving object management systems manage spatiotemporal data, which change their location over time such as people, animals, and cars. These moving object management systems can be applied to vehicle location tracking, digital battlefield, location-based service, and so on. The existing moving object management systems only manage past or future location of the moving objects separately. Therefore, they cannot suggest estimation method of uncertain past or future location of the moving objects. In this paper, we propose a moving object management system, which not only manages historical data of the moving objects, but also predicts past and future location of the moving objects using historical data stored in database. We define the moving objects for vehicle location tracking and propose a moving object database structure. Finally, we suggest an execution model of the proposed system and apply the execution model to a virtual scenario for vehicle tracking.

키워드 : 차량 위치 추적(Vehicle Location Tracking), 이동 객체 관리 시스템(Moving Object Management System), 불확실한 위치 추정(Prediction of Uncertain Location)

1. 서 론

차량 위치 추적 시스템은 GPS, 비콘, 루프 검지기 등의 차량 위치 탐지기로부터 전송받은 위치 정보를 토대로, 현재 차량의 위치 정보나 교통량의 흐름 등을 실시간으로 제공해주는 시스템이다[1-4]. 특히 이동 차량은 시간의 변화에 따라 그 위치가 연속적으로 변경되는 이동 객체의 특성을 가지므로, 이동 객체를 효율적으로 저장 및 관리하기 위한 연구가 필요하게 되었다. 지금까지 이동 객체의 연구는 데이터 모델

링, 질의 표현, 인덱스 구조, 불확실성 처리 방법 등으로 세분화되어 연구되었으며, 차량 위치 추적을 위한 관련 응용 시스템들이 개발되었다.

그런데 차량 위치 추적을 위해 지금까지 연구된 이동 객체 관리 시스템들은 이동 객체의 과거 및 미래의 위치정보를 개별적으로 관리하고 있다. 이로 인해 하나의 데이터베이스에서 과거 및 미래의 위치 정보를 동시에 관리하지 못하며, 동일한 데이터베이스에 저장된 이력 위치정보를 이용하여 이동 객체의 불확실한 과거 및 미래의 위치 추정 방법을 제시하지 못하고 있다.

따라서 이 논문에서는 이와 같은 문제점을 해결할 수 있는 이동 객체 관리 시스템을 제안한다. 이를 위해 먼저 이동 객체 관리 시스템에 적용될 차량 위치 정보를 모델링 한다. 그

* 이 연구는 ETRI의 우정연구지원과 한국과학재단 특정기초연구(R01-1999-00234)의 연구비 지원으로 수행되었음.

† 준 회원 : 충북대학교 대학원 전자계산학과

†† 정 회원 : 한국전자통신연구원 우정기술연구소 선임연구원

††† 총신회원 : 충북대학교 전기전자및컴퓨터공학부 컴퓨터정보통신 연구소 교수
논문접수 : 2002년 6월 5일, 심사완료 : 2002년 7월 25일

리고 이동 객체 연산 처리기, 불확실한 위치 처리기, 실시간 차량 위치정보 수신 및 저장기, 이동 객체 데이터베이스로 구성된 이동 객체 관리 시스템 구조를 제시한 후, 구성요소별 기능 및 처리 알고리즘을 제시한다. 특히 제안하는 이동 객체 관리 시스템은 데이터베이스에 저장된 이력 위치정보를 토대로 선형 보간법을 이용한 과거 위치 추정 방법과 이력 위치정보의 평균 이동 값을 이용한 미래의 위치 추정 방법을 제시한다.

제안 시스템은 다음과 같은 특징을 가진다. 첫째, 기존에 개별적으로 관리되던 이동 객체의 과거 및 미래의 위치정보를 하나의 데이터베이스를 이용하여 동시에 관리한다. 둘째, 데이터베이스에 저장되지 않은 과거의 위치 정보를 추정한다. 셋째, 차량 위치 탐지기의 통신 및 기타 장애로 인해 위치 정보를 제공받지 못할 경우 가까운 미래의 위치를 추정한다.

이 논문의 구성 및 내용 전개는 다음과 같다. 먼저 2절에서는 이동 객체 관리 시스템의 관련 연구를 서술한다. 3절에서는 이동 차량의 위치정보 모델링을 위한 이동 객체 및 연산자를 정의하고, 데이터베이스 구조를 제시한다. 4절에서는 차량 위치 추적을 위한 이동 객체 관리 시스템의 구조 및 모듈별 처리 알고리즘을 기술한다. 5절에서는 제안한 이동 객체 관리 시스템의 실행 모델 및 가상 시나리오를 이용한 구현 결과를 예시하고, 마지막으로 6절에서는 결론을 내린다.

2. 관련 연구

이동 객체는 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 객체에는 사람, 동물, 자동차, 비행기, 배 등이 있다. 이동 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 같은 이동 객체를 저장 및 관리하는 이동 객체 데이터베이스는 위치 기반 서비스, 차량 위치 추적, 물류 관리 등을 위한 기본 소프트웨어 인프라로 인식되고 있다[5-9].

이동 객체 관리를 위한 대표적인 응용 시스템 연구에는 DOMINO, CHOROCHRONOS, DEDALE, Battlefield Analysis가 있다. DOMINO[9-13]는 이동 객체의 수송에 대한 웹 기반 실시간 케도 응용의 개발을 촉진하는 이동 객체 소프트웨어 도구로서, DBMS 기술을 활용하여 만든 실시간 위치 추적 시스템 프로토타입이다. 그러나 DOMINO 프로토타입은 이동 객체의 현재 위치, 속도, 방향정보를 이용하여 미래의 이동 위치를 예측하는 방법에 주로 초점을 맞추고 있다. 따라서 과거 시점을 포함하는 이동 객체의 완전한 이동 경로인 궤적(trajecory)을 관리할 수 없는 단점을 가진다.

CHOROCHRONOS[14-18]는 시공간 데이터베이스의 특수한 형태인 이동 객체에 관한 연구가 집중적으로 수행되었고, 이동 객체의 데이터 모델링 및 인덱싱에 관한 연구 결과가 가장 많이 발표하였다. 특히 이 연구는 GPS 기반의 수송 관리 시스템과 멀티미디어 시스템에 적용한 응용 시나리오를 제시하였다. 그러나 아직 이동 객체 데이터베이스를 활용한 응용 시스템의 모델 및 개발 사례는 제시되지 않고 있으며, 개발 중인 질의처리 시스템에서는 이동 객체의 불확실한 과거 및 미래의 위치 정보 추정에 관한 방법이 제시되지 않고 있다.

DEDALE[19, 20]은 제약사항 데이터베이스 모델을 이용하여 시공간 데이터를 모델링하고 질의처리를 하기 위해 개발된 프로토타입으로, 기존의 시공간 데이터의 모델뿐만 아니라 이동 객체의 궤적 등과 같은 데이터 모델 및 질의 표현도 제공하였다. DEDALE은 고급 수준의 개발 및 최적화를 위한 질의어의 확장을 위해, 공간 질의에 대한 최적 계산 기법의 사용 대신 기하 데이터에 대해 선형 제약사항 추상화를 제공하였다. 그러나 DEDALE 프로토타입은 데이터베이스에 시간의 변화에 따른 이동 객체의(x, y) 좌표 값이 직접 저장되지 않고, 특정 구간의 궤적을 표현하는 선형 제약사항(linear constraint)의 공식이 저장된다. 이로 인해 빈번하게 이동하는 객체의 실시간 위치 추적을 위한 응용 시스템 개발에는 부적합한 특징을 가진다.

Battlefield Analysis[21-23]는 모의 전장에서 이동하는 부대 및 탱크들의 움직임을 예측하여 이를 의사결정에 활용할 수 있도록 개발된 전장분석 프로토타입이다. 전장분석 프로토타입은 이동 객체 관리기와 추론 엔진을 접목시키고자 하는데 초점이 맞추어졌다. 특히, 시공간 이동 객체의 연산 결과를 추론 엔진에서 활용하는 새로운 이동 객체 추론 모델을 제시하였다. 그러나 이동 객체 관리 시스템의 실시간 환경이 고려되지 않았으며, 특정 도메인 지식을 활용한 이동 객체의 미래의 위치정보만을 예측하고 있다. 따라서 임의의 과거 및 미래 시점에 대한 모든 위치정보를 제공하지 못하고 있다.

지금까지의 관련 연구를 분석해보면 이동 객체의 과거 및 미래의 위치정보를 하나의 데이터베이스에서 동시에 관리하지 못하며, 동일한 데이터베이스에 저장된 이력 위치정보를 이용하여 이동 객체의 불확실한 과거 및 미래의 위치 추정 방법을 제시하지 못하는 문제점을 가지고 있다.

3. 이동 차량의 위치 정보 모델링

이 절에서는 차량 위치 추적을 위한 이동 객체 관리 시스템에 적용될 이동 차량의 위치정보를 모델링 한다. 이를 위해 이동 점만을 대상으로 하는 이동 객체 데이터와 연산자를 정의하고, 이동 객체의 이력 정보 및 현재 정보를 데이터베이스에 저장하는 형태를 제시한다.

3.1 이동 객체 정의

이 논문에서는 이동 차량의 위치정보 관리를 위해 이동 객체를 대상으로 하는 이동 객체만을 고려한 이동 차량 데이터를 다음과 같이 정의한다.

[정의 1] 이동 객체

시간의 변화에 따라 객체의 위치 값만 변화되는 이동 객체를 말한다. 이동 객체 MP 는 시간 속성, 공간 속성, 일반 속성을 가지며, $MP = \langle T_A, S_A, G_A \rangle$ 가 된다. □

[정의 2] 시간 속성

MP 의 시간 속성 $T_A = \langle vt_s, vt_e \rangle$ 로 구성되며 vt_s 는 시작 시간, vt_e 는 종료 시간을 나타낸다. 이 때 vt_s 와 vt_e 는 유효시간의 집합 S_{VT} 의 원소가 된다. 유효시간은 실제 세계에서 발생된 시간을 나타내며, $S_{VT} = \{t_0, t_1, t_2, \dots, t_k, \dots, t_{now}\}$ 이고, 각 원소들은 $t_0 < t_1 < t_2 < \dots < t_k < \dots < t_{now}$ 의 순서를 가진다. $t_k = t_{k-1} + 1$, $t_k = t_0 + k$, $k \geq 0$ 인 정수로 정의된다. t_{now} 는 현재 시간을 의미하는 시간 상수이다. □

유효시간의 도메인은 선형 시간(linear time), 이산 시간(discrete time), 절대 시간(absolute time)이며, 하나의 이동 객체 데이터베이스는 동일한 유효시간의 주기(granularity)를 가진다.

[정의 3] 공간 속성

MP 의 공간 속성 $S_A = \langle x, y \rangle$, $x, y \in R$, R 은 실수이다. □

[정의 4] 이동 객체 데이터베이스

이동 객체 데이터베이스를 구성하는 이동 객체들의 집합은 $S_{MP} = \{MP_0, MP_1, \dots, MP_n\}$ 이다. S_{MP} 로 구성된 이동 객체 데이터베이스의 이력 집합은 $H_{MP} = \{H_{MP_0}, H_{MP_1}, \dots, H_{MP_n}\}$ 이다. S_{MP} 에 속하는 각각의 MP_i 에 대한 모든 이력 집합은 $H_{MP_i} = \{mp_{i_0}, mp_{i_1}, \dots, mp_{i_k}\}$ 이고, $mp_{i_k} = \langle T_A(mp_{i_k}), S_A(mp_{i_k}), G_A(mp_{i_k}) \rangle$ 이다. □

[정의 4]에서 mp_{i_k} 는 MP_i 의 k 번째 이력 정보를 의미하고, $T_A(mp_{i_k})$ 는 MP_i 의 k 번째 시간 속성, $S_A(mp_{i_k})$ 는 k 번째 공간 속성, $G_A(mp_{i_k})$ 는 k 번째 일반 속성을 의미한다.

3.2 연산자

이동 차량의 위치정보 관리를 위해 다음과 같은 이동 객체 연산자를 정의한다. 이동 객체의 일반 속성에 대한 검색은 기존의 상용 데이터베이스 시스템이 제공하는 기능을 그대로 사용하므로 별도로 정의하지 않는다. 이동 객체의 위치정보 관련 연산자로 $m_distance$, $trajectory$, $length$, $position_at$, $m_nearest$, $m_farthest$ 를 <표 1>과 같이 정의한다.

<표 1>은 기존의 이동 객체 연구[5-8]에서 제시된 연산자들을 토대로 기본 연산자만을 정의한 것이다. 입력 및 출력 값에서 G_A 는 차량 소유주, 운전자, 차종 등과 같은 이동 객체의 일반 속성을 의미한다. S_A 는 공간 속성을 나타내고, T_A 는 시간 속성인 유효시간을 의미한다. $Real$ 은 실수 값을 나타내며, $\{ \}$ 로 묶인 속성들은 집합을 의미한다.

3.3 데이터베이스 구조

3.1절에서 정의한 이동 객체를 관리하기 위한 데이터 저장 구조는 관계 데이터베이스를 기반으로 하며, 다음과 같이 3개의 릴레이션으로 구분하여 저장한다. 먼저, 객체 릴레이션 FleetObject에는 이동 객체의 일반 속성정보가 저장된다.

<표 2> FleetObject : 객체의 일반 속성정보 저장 릴레이션

CarId	Name	Type	...
356583455	fleet_1	truck	...
356583466	fleet_2	container	...

<표 2>의 FleetObject 릴레이션에서 CarId는 객체의 식별자로서 키 값이 된다. 기타 일반 속성 정보는 객체의 이름을 나타내는 Name이나, 객체의 유형을 나타내는 Type 등을 가질 수 있다. 현재 릴레이션 FleetCurrent에는 이동 객체의 현재의 시간 및 공간 속성정보가 저장된다.

<표 3> FleetCurrent : 현재의 위치정보 저장 릴레이션

CarId	Time	X	Y
356583455	2002-03-01-08-20-00	202053.00	444755.69
356583466	2002-03-01-09-00-00	201153.42	445201.23

<표 3>의 FleetCurrent 릴레이션은 <표 4>의 Fleet-

<표 1> 이동 객체 연산자

종 류	입력 값	출력 값	기 능
$m_distance$	$G_A \times G_A \times T_A$	$Real$	특정 유효시간 동안 두 이동 객체간의 거리 계산
$trajectory$	$G_A \times T_A$	$\{S_A\}$	이동 객체의 특정 유효시간 동안의 이동 경로 추출
$length$	$\{S_A\}$	$Real$	이동 객체의 특정 유효시간 동안의 이동 거리 계산
$position_at$	$G_A \times T_A$	S_A	입력의 한 시점에 대한 이동 객체의 위치 검색
$m_nearest$	$G_A \times T_A$	S_A	특정 유효시간 동안 가장 가까이 위치하는 객체 검색
$m_farthest$	$G_A \times T_A$	S_A	특정 유효시간 동안 가장 멀리 존재하는 객체 검색

History 릴레이션과 구조가 동일하다. 가장 최근에 저장된 이동 객체의 위치 좌표만을 FleetCurrent 릴레이션에 저장하고, 그 이전의 위치 정보는 모두 이력 릴레이션에 저장한다.

<표 4> FleetHistory : 이력 위치정보 저장 릴레이션

CarId	Time	X	Y
356583455	2002-03-01-07-50-00	200998.11	445124.01
356583455	2002-03-01-07-55-00	201287.75	445238.44
356583455	2002-03-01-08-00-00	201566.67	445345.72
356583455	2002-03-01-08-05-00	201809.84	445410.08
356583455	2002-03-01-08-10-00	201888.51	445188.38
356583455	2002-03-01-08-15-00	201974.33	444973.82
356583455	2002-03-01-08-20-00	202053.00	444755.69
356583466	2002-03-01-09-00-00	201153.42	445201.23

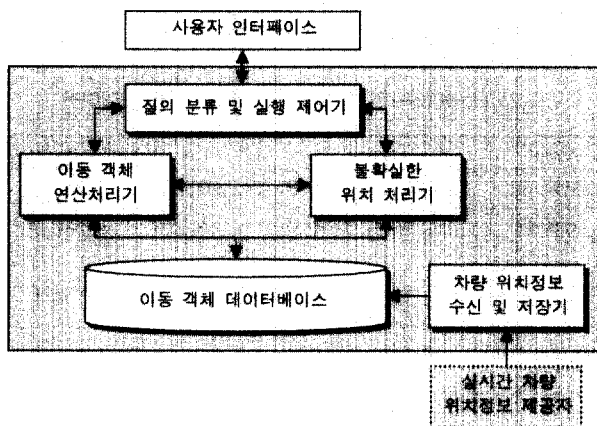
<표 4>의 FleetHistory 릴레이션에는 이동 객체 데이터베이스에 저장된 모든 과거 시점의 이력 위치 정보를 저장한다. <표 4>는 5분 간격으로 차량의 위치정보를 저장한 예이다.

4. 이동 객체 관리 시스템의 설계

이 절에서는 차량 위치 추적을 위한 이동 객체 관리 시스템 구조 및 기능을 제시한다. 그리고 각 구성요소별 처리 알고리즘을 기술한다.

4.1 시스템 구조

제안 시스템은 이동 객체 연산처리기, 불확실한 위치 처리기, 차량 위치정보 수신 및 저장기, 이동 객체 데이터베이스로 구성되며, (그림 1)과 같은 구조를 가진다.



(그림 1) 이동 객체 관리 시스템의 구조

(그림 1)의 이동 객체 관리 시스템의 각 구성요소별 기능은 다음과 같다.

4.1.1 질의 분류 및 실행 제어기

질의 분류 및 실행 제어기는 사용자 인터페이스를 통해 입

력된 이동 객체 질의처리를 위한 실행 연산자의 호출 및 제어 기능을 가진다. 이를 위해 질의 수행 계획을 수립하여 해당 이동 객체 연산자의 순차적인 실행을 명령하고, 각 연산자들의 수행 결과를 통합하여 최종 질의 처리 결과를 생성한 후 사용자가 원하는 형태로 출력한다.

4.1.2 이동 객체 연산 처리기

이동 객체 연산 처리기는 모든 질의 수행을 위한 실행 연산자를 처리하는 부분이다. 시스템에서 처리되는 이동 객체 연산자의 종류는 3.2절에서 정의한 이동 객체 데이터의 조작을 위한 기본 연산자이다. 이동 객체 연산 처리기는 데이터베이스에 저장된 위치정보를 토대로 실행된다. 이로 인해 이동 객체 데이터베이스에 저장되지 않은 과거 및 미래의 특정 시간의 위치정보에 관한 질의가 요청된 경우에는 자체적으로 처리할 수가 없다. 따라서 이런 경우에는 불확실한 위치 처리기를 통해 계산된 위치 값을 가지고 연산처리를 한다.

4.1.3 불확실한 위치 처리기

불확실한 위치 처리기는 이동 객체 데이터베이스에 저장되지 않은 특정한 과거 및 미래의 시간에 대한 객체의 불확실한 위치 값을 계산하여 결과를 반환하는 기능을 가진다. 이를 처리하기 위해 시간의 변화에 따라 변경되는 이동 객체의 위치 변화 함수를 생성한다. 위치 변화 함수에는 과거의 위치를 추정하는 함수와 미래의 위치 추정 함수가 사용된다.

4.1.4 차량 위치정보 수신 및 저장기

차량 위치정보 수신 및 저장기는 이동 차량의 실시간 위치 정보를 수신하고 이를 이동 객체 데이터베이스에 저장한다. 이동 객체의 실시간 위치 정보는 외부 데이터 제공자로부터 일정한 시간의 주기마다 전송된다. 실시간 데이터 제공자는 이동 차량의 위치 정보를 수신하여 일정한 형태의 패킷으로 변환한 후 이를 이동 객체 관리 시스템에 전송한다.

4.1.5 이동 객체 데이터베이스

이동 객체 데이터베이스에는 이동 차량의 속성정보 및 위치정보를 저장한다. 과거의 이력정보는 특정 시간의 간격에 따라 이력 테이블에 저장된다. 그리고 실시간 위치정보는 현재 테이블에 따로 저장된다. 미래의 위치정보는 데이터베이스에 저장되지 않고 연산 처리를 위해 필요한 경우 불확실한 위치 처리기를 통해 계산된다. 이동 객체 데이터베이스는 기존의 상용 데이터베이스 시스템을 이용한다.

4.2 이동 객체 연산처리기

이동 객체 연산 처리기는 사용자 질의를 유형별로 분류하여 해당 질의의 실행 모듈을 호출한다. 실행 연산자의 종류는 3.2절의 <표 1>에서 정의한 것과 같다. 각 연산처리 모듈에서는 과거 및 미래의 시점에 대한 질의가 입력된 경우에 불확실한 위치 처리기를 호출하여 결과를 구한다.

```

Algorithm position_at ( name, time)
입력 => name : 임의의 이동 객체 MP의 이름, time : 검색하고 싶은
질의 시점
출력 => x : time 시점의 x 좌표 값, y : time 시점의 y 좌표 값
Begin
FleetObject에서 입력된 name을 가지는 객체의 CarId를 검색
If (검색 결과가 null이 아니면) Then
current_time ← FleetCurrent에서 CarId를 갖는 객체의 time 속성 검색
If (time = current_time) Then
x ← FleetCurrent에서 CarId를 갖는 객체의 x 속성 검색
y ← FleetCurrent에서 CarId를 갖는 객체의 y 속성 검색
Else If (time < current_time) Then
x ← FleetHistory에서 CarId와 time 속성을 갖는 튜플의 x 값
검색
y ← FleetHistory에서 CarId와 time 속성을 갖는 튜플의 y 값
검색
If (검색 결과가 null 이면) Then
past_location( CarId, time) 모듈 호출한 후 x, y 값 구함
// past_location : 과거 시점의 불확실한 위치 추정 모듈
Else future_location( CarId, time) 모듈 호출한 후 x, y 값 구함
// future_location : 미래 시점의 불확실한 위치 추정 모듈
return x and y // 이동 객체의 time 시점의 x, y 좌표 값 반환
End
    
```

(그림 2) 이동 객체 연산 처리 알고리즘

(그림 2)는 특정 시점에 대한 이동 객체의 위치를 검색하는 position_at 연산자의 실행 알고리즘이다. 알고리즘에 사용되는 데이터베이스는 3.3절에서 정의한 FleetObject, FleetCurrent, FleetHistory 릴레이션을 토대로 한다. FleetObject에 존재하는 이동 객체를 대상으로, 입력된 시간 값을 비교하여 과거, 현재, 또는 미래의 위치 값을 구한다.

4.3 불확실한 위치 처리기

과거 및 미래의 위치 추정기는 이동 객체 데이터베이스에 저장되지 않은 특정한 시간에 대한 객체의 불확실한 위치 값을 계산하여 결과를 반환하는 기능을 가진다. 이 모듈에서는 과거의 위치정보 추정을 위해 선형 보간법을 이용한다. 그리고 미래의 위치정보 추정을 위해서 이력 테이블에 저장된 위치정보의 평균 이동정보를 활용한다.

4.3.1 과거의 위치 추정 모듈

이력 데이터베이스에 저장되는 시점이 $\{t_0, t_1, t_2, \dots, t_{i-1}, t_i, t_{i+1}, \dots, t_{now-1}, t_{now}\}$ 일 때, $t_i < t_p < t_{i+1}$ 의 조건을 만족하는 시점 t_p 에 대해 (t_i, x_{t_i}) 와 $(t_{i+1}, x_{t_{i+1}})$ 의 위치좌표 쌍을 이용하여 $x(t_p)$ 함수를 구하고, (t_i, y_{t_i}) 와 $(t_{i+1}, y_{t_{i+1}})$ 의 위치좌표 쌍을 이용하여 $y(t_p)$ 함수를 구한다. 이 때 함수 $x(t_p)$ 와 $y(t_p)$ 를 이용하여 과거의 시점 t_p 의 위치 값을 추정한다.

(그림 3)의 알고리즘 past_location은 이동 객체 MP의 CarId와 과거의 임의의 시점 t_p 를 입력 값으로 받는다. 이후 객체의 속성 정보가 저장된 FleetObject 릴레이션에서 입력된 CarId를 가지는 객체를 검색한다. 검색된 객체가 존재할 경

우, 이력 정보가 저장된 FleetHistory 릴레이션에서 t_p 와 가장 인접한 이전의 시점 t_i 의 (x, y) 좌표 쌍 $(t_i, x_{t_i}), (t_i, y_{t_i})$ 를 검색한다. 그리고 t_p 와 가장 인접한 이후의 시점 t_{i+1} 의 (x, y) 좌표 쌍 $(t_{i+1}, x_{t_{i+1}}), (t_{i+1}, y_{t_{i+1}})$ 를 검색한다. 검색된 위치 좌표를 이용하여 x_{t_p} 와 y_{t_p} 값을 구하여 결과를 반환한다.

```

Algorithm past_location ( CarId, t_p )
입력 => CarId : 임의의 이동 객체 MP의 식별자, t_p : 과거의 특정 시점
출력 => x_t : 시점 t_p시 MP의 x 좌표 값, y_t : 시점 t_p시 MP의 y 좌표 값
Begin
FleetObject에서 입력된 CarId를 가지는 객체 검색
If (검색 결과가 null이 아니면) Then
FleetHistory에서  $t_i < t_p < t_{i+1}$ 의 조건을 만족하는 위치 좌표 쌍
 $(t_i, x_{t_i}), (t_i, y_{t_i})$ 와  $(t_{i+1}, x_{t_{i+1}}), (t_{i+1}, y_{t_{i+1}})$ 을 검색
If (검색 결과가 null이 아니면) Then

$$x_{t_p} \leftarrow \frac{x_{t_{i+1}} - x_{t_i}}{t_{i+1} - t_i} (t_p - t_{i+1}) + x_{t_{i+1}}$$


$$y_{t_p} \leftarrow \frac{y_{t_{i+1}} - y_{t_i}}{t_{i+1} - t_i} (t_p - t_{i+1}) + y_{t_{i+1}}$$

Else  $x_{t_p} \leftarrow$  오류 값,  $y_{t_p} \leftarrow$  오류 값
Return  $x_{t_p}$  and  $y_{t_p}$  // 과거의  $t_p$  시점의  $(x, y)$  좌표 값 반환
End
    
```

(그림 3) 과거의 위치 추정 알고리즘

4.3.2 미래의 위치 추정 모듈

미래의 위치 변화 함수를 생성하기 위해 과거 이력정보의 평균 이동 정보를 활용한 모델을 제시한다. 입력된 미래의 임의의 시점 t_f 가 $t_{now} < t_f$ 의 조건을 만족할 경우, t_{now} 이전의 N 개의 이력 위치 좌표를 검색하여 함수 $x(t_f)$ 와 $y(t_f)$ 를 구하여 미래의 시점 t_f 의 위치 값을 추정한다.

(그림 4)의 미래의 위치 추정 함수에 사용되는 과거 이력 위치 정보의 개수는 N 이라고 가정한다. 알고리즘 future_location은 이동 객체 MP의 CarId와 미래의 임의의 시점 t_f 를 입력 값으로 받은 후, FleetObject 릴레이션에서 입력된 CarId를 가지는 객체를 검색한다. 검색된 객체가 존재할 경우, 이력 정보가 저장된 FleetHistory 릴레이션에서 t_{now} 시점 이전의 N 개의 위치 좌표를 검색하여 각각 x_i, y_i, t_i 의 배열에 저장한다. 이 때 배열의 첨자 i 는 1~ N 까지의 값을 가진다. 임시 배열에 저장된 위치 좌표를 이용하여 x_{t_f} 와 y_{t_f} 값을 구하여 결과를 반환한다. 이 때 x_{t_f} 와 y_{t_f} 를 구하는 수식의 설명은 다음과 같다. 먼저, $\sum_{i=2}^N (x_{t_i} - x_{t_{i-1}})$ 와 $\sum_{i=2}^N (y_{t_i} - y_{t_{i-1}})$ 는 N 개의 이력 위치 정보의 x 축과 y 축의 이동거리의 합이다. $\sum_{i=2}^N (t_i - t_{i-1})$ 는 N 개의 이력 위치 정보의 이동시간의 합이다. $(t_f - t_N)$ 은 이동 객체 MP가 현재 시점 t_N 에서 미래 시점

t_f 까지 이동할 향후 이동시간이다. x_{t_N} 과 y_{t_N} 은 t_N 시점의 x, y 좌표 값을 나타낸다.

```

Algorithm future_location ( CarId, t_f )
입력 => CarId : 임의의 이동 객체 MP의 식별자, t_f : 미래의 특정
시점
출력 => x_t : 시점 t_f 시 MP의 x 좌표 값, y_t : 시점 t_f 시 MP의 y 좌
표 값
N : 미래의 위치 좌표 추정에 사용될 최근 이동정보의 위치 좌표
쌍의 개수
Begin
FleetObject에서 입력된 CarId를 가지는 객체 검색
If (검색 결과가 null이 아니면) Then
FleetHistory에서 t_now 이전의 최근 N 개의 위치 좌표 쌍을 검색
If (검색 결과가 null이 아니면) Then
검색된 위치 좌표를 임시 배열 x_i, y_i, t_i 에 저장 (i=1~N)

x_t ← ( ∑_{i=2}^N (x_i - x_{t_{i-1}}) ÷ ∑_{i=2}^N (t_i - t_{i-1})) × (t_f - t_N) + x_{t_N}
y_t ← ( ∑_{i=2}^N (y_i - y_{t_{i-1}}) ÷ ∑_{i=2}^N (t_i - t_{i-1})) × (t_f - t_N) + y_{t_N}

Else x_t ← 오류 값, y_t ← 오류 값
Return x_t, and y_t, // 미래의 t_f 시점의 (x, y) 좌표 값 반환
End
    
```

(그림 4) 미래의 위치 추정 알고리즘

4.4 실시간 위치정보 수신 및 저장기

외부 데이터 제공자로부터 전송되는 이동 객체의 위치 정보는 <표 5>와 같은 형태의 패킷으로 구성된다. <표 5>는 이 논문에서 설계한 이동 객체 데이터베이스에 저장될 위치 정보만을 구성요소로 하여 작성한 패킷구조이다. 만약 이동 차량의 다른 정보들을 필요로 할 경우에는 추가적인 정보를 패킷의 구성요소에 넣을 수 있다. 그러나 통신비용 및 데이터 송수신 속도를 고려하여 이 논문에서는 최소한의 위치정보만을 이용한다.

<표 5> 실시간 차량 위치 정보 패킷

구성 요소	크기(바이트)	설 명
Header	1	데이터 전송 헤더
Size	4	Size 이후의 데이터 길이
Code	1	메시지 구분 코드
CarId	4	차량 구분 ID
X	4	차량의 현재 위치 X 좌표
Y	4	차량의 현재 위치 Y 좌표
Time	3	현재 시간(시/분/초)
Validity	1	데이터 유효성 여부
Dummy	9	미결정 및 추가 데이터 입력 부분

이동 차량의 실시간 이동정보는 <표 5>와 같은 구조로 일정한 시간 주기에 따라 데이터 로더에 전송된다. 데이터 로더

는 수신된 위치정보를 (그림 5)와 같은 데이터 변환 및 저장 알고리즘을 통해 이동 객체 데이터베이스에 저장한다.

```

Algorithm data_translator ( packet )
입력 => packet : 버퍼 형태로 구성된 패킷 데이터 배열
Begin
CarId ← packet 배열 중 CarId 부분인 4번째 값을 정수로 변환
x ← packet 배열 중 x 좌표 값을 나타내는 5번째 값을 실수로 변환
y ← packet 배열 중 y 좌표 값을 나타내는 6번째 값을 실수로 변환
time ← packet 배열 중 time 값을 나타내는 7번째 값을 문자형으로
변환
data_store ( CarId, time, x, y ) 모듈을 호출하여 실행
End
Algorithm data_store ( CarId, time, x, y )
입력 => CarId : 차량 식별자, time : 시간 값, x : x 좌표 값, y : y 좌표 값
Begin
FleetHistory에 CarId, time, x, y를 가지는 새로운 튜플 삽입
FleetCurrent에서 CarId를 가지는 객체 검색
If (검색 결과가 null이 아니면) Then
CarId를 가지는 튜플의 time, x, y 속성 값을 갱신
Else
FleetCurrent에 CarId, time, x, y를 가지는 새로운 튜플 삽입
End
    
```

(그림 5) 위치정보 변환 및 저장 알고리즘

(그림 5)에서 data_translator는 데이터 로더를 통해 수신된 위치정보 패킷 중에서 이동 객체 데이터베이스에 실제 저장될 CarID, X, Y, Time 부분의 값을 해당 자료형으로 변환한다. 그리고 변환된 데이터는 data_store 모듈을 통해 이동 객체 데이터베이스에 저장한다. data_store에서는 FleetHistory 릴레이션에 새로운 CarID, X, Y, Time 값을 가지는 튜플을 삽입한다. 만약, FleetCurrent 릴레이션에 CarId를 가지는 튜플이 존재하면 X, Y, Time 속성을 갱신하고, 그렇지 않으면 CarID, X, Y, Time 값을 가지는 새로운 튜플을 삽입한다.

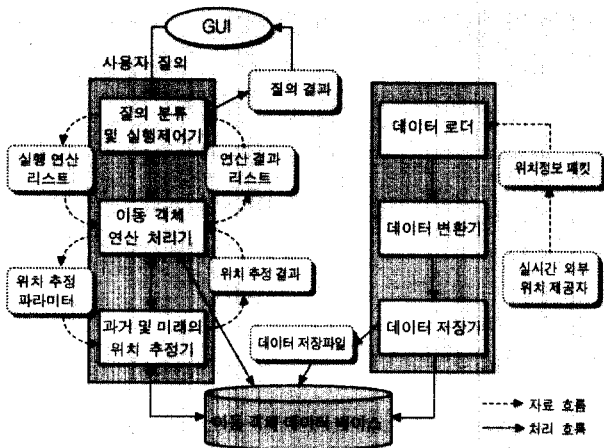
5. 제안 시스템의 실행 모델 및 적용

이 절에서는 제안 시스템의 전체적인 동작 과정을 예시한다. 그리고 제안 시스템을 가상 시나리오에 적용하여 구현한 차량 추적 실시간 이동 객체 관리 시스템의 실행 결과를 보인다.

5.1 시스템의 동작

제안한 이동 객체 관리 시스템의 전체적인 동작은 (그림 6)과 같이 질의 수행 및 불확실한 위치 처리부와 데이터 수신 및 저장부의 두 부분으로 나누어 실행된다. 질의 수행 및 불확실한 위치 처리 부분에서는 사용자 질의를 입력받아 이동 객체 연산 처리를 한 후 그 결과를 다시 사용자에게 반환한다. 데이터 수신 및 저장 부분에서는 외부에서 송신되는 실시간 차량 위치정보를 수신하고 이를 데이터베이스에 저장한다.

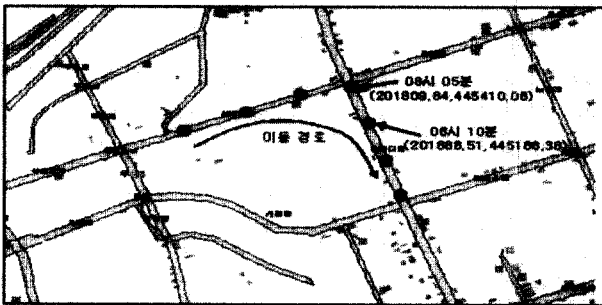
(그림 6)의 실행 모델을 예를 들어 설명하면 다음과 같다.



<질의 수행 및 불확실한 위치 처리 부분> <데이터 수신 및 저장 부분>

(그림 6) 이동 객체 관리 시스템의 동작 과정

먼저, 특정 차량이 (그림 7)과 같이 서울 시내의 '신반포로'에서 '강남대로' 쪽으로 이동하고 있고, 이 때 차량의 위치정보는 5분 간격으로 데이터 로더를 통해 수신된다고 가정하자. 수신된 위치정보는 4.4절의 알고리즘 (그림 5)와 같이 처리된 후, 3.3절에서 제시한 <표 2>, <표 3>, <표 4>와 같이 Fleet-Object, FleetCurrent, FleetHistory 릴레이션에 순서대로 저장된다. 이와 같은 차량의 이동 과정을 지도 위에 표현하면 다음과 같다.



(그림 7) 차량 이동 과정 예

(그림 7)과 같이 이동하는 차량의 위치정보를 토대로 질의 1이 처리되는 과정을 살펴본다.

- 질의 1 : "2002년 1월 1일 08시 08분에 차량명이 'fleet_1' 인 차량의 위치를 검색하시오."

사용자 질의 1이 입력되면 (그림 6)의 실행 모델에서와 같이 첫 번째로 질의 분류기가 수행된다. 현재 입력된 질의는 임의의 한 시점에 대한 이동 객체의 위치를 검색하는 질의이므로, 3.2절의 이동 객체 연산자 중 *position_at* 연산자가 추출된다. 두 번째로, 이동 객체 연산 처리기가 수행된다. *position_at* 연산자는 4.2절의 (그림 2)에서 기술한 알고리즘을 통해 처리된다. 이 때 FleetObject 릴레이션에서 Name이 'fleet_1'인 객체의 CarId '356583455'가 검색되고, FleetCurrent와 Fleet-

History 릴레이션에서 2002년 1월 1일 08시 08분에 해당되는 (x, y) 좌표 값을 검색한다. 그러나 <표 4>와 (그림 7)에서 볼 수 있듯이, 08시 08분에 해당되는 차량의 위치 정보는 데이터베이스에 저장되지 않았다. 따라서 세 번째로, 불확실한 위치 처리기가 수행되는데, 08시 08분의 질의 시점이 과거이므로 과거 위치 추정 모듈이 실행된다. 4.3.1절 (그림 3)의 알고리즘에 의해 과거 위치 추정 모듈이 실행된다.

<표 6> 과거 위치 추정을 위해 검색된 위치 정보

Time	X	Y
2002-03-01-08-05-00	201809.84	445410.08
2002-03-01-08-10-00	201888.51	445188.38

<표 6>은 과거 위치 추정을 위해 FleetHistory 릴레이션에서 검색된 08시 05분의 위치 좌표 (201809.84, 445410.08)와 08시 10분의 위치 좌표(201888.51, 445188.38)이다. 이 두 위치정보를 사용하여 4.3.1절 (그림 3)의 과거 위치 추정 연산 결과로 생성된 08시 08분의 위치 좌표는(201857.042, 445277.06)이 된다. 다음 질의 2의 경우를 살펴보자.

- 질의 2 : "2002년 1월 1일 08시 23분에 차량명이 'fleet_1' 인 차량의 위치를 검색하시오."

질의 2의 수행 과정은 질의 1과 동일하고, 데이터베이스에 저장되지 않은 미래의 질의 시점에 대한 연산 처리가 요구된다. 따라서 4.3.2절 (그림 4)의 알고리즘에 의해 미래 위치 추정 모듈이 실행된다. 이 때 미래 위치 추정에 사용되는 이력 위치좌표 쌍의 개수를 나타내는 N의 값이 4라고 가정하자.

<표 7> 미래 위치 추정을 위해 검색된 위치 정보

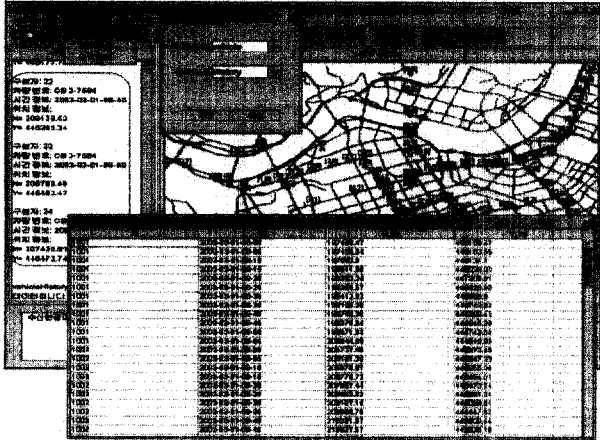
Time	X	Y
2002-03-01-08-05-00	201809.84	445410.08
2002-03-01-08-10-00	201888.51	445188.38
2002-03-01-08-15-00	201974.33	444973.82
2002-03-01-08-20-00	202053.00	444756.69

<표 7>은 미래 위치 추정을 위해 FleetHistory 릴레이션에서 검색된 네 개의 이력 위치정보이다. 이 위치 정보를 이용하여 4.3.2절 (그림 4)의 미래 위치 추정 연산 결과로 생성된 08시 23분의 차량 위치 좌표는 (202101.632, 444624.821)이 된다.

5.2 가상 시나리오 및 구현

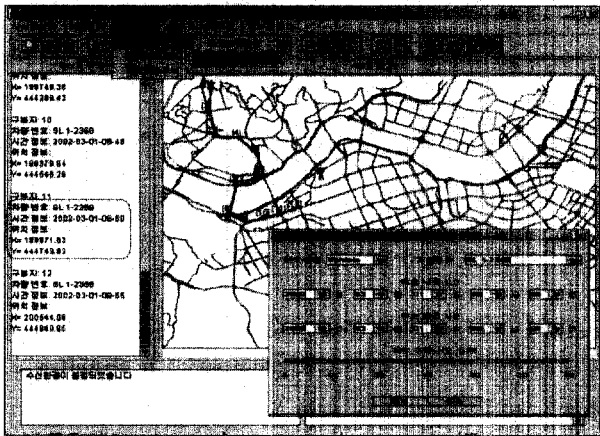
제안한 이동 객체 관리 시스템 구조 및 알고리즘을 가상 시나리오에 적용하여 구현하였다. 먼저, 서울시 주요 간선도로에서 차량이 이동하고 있을 때, 5분 간격으로 이동 차량의 위치 정보를 수신 받아 데이터베이스에 저장함을 가정한다. 데이터 생성을 위해 ZEUS 공간 객체 관리 시스템에 구축된 서울시 데이터베이스의 맵 정보와 좌표를 활용하였다. 사용

된 위치 좌표는 TM 좌표 체계를 따른 것이다. 이동 차량의 위치 정보는 2개의 객체에 대해 약 150여개의 이력 튜플을 저장하였다. 구현은 Windows98/NT/2000 운영체제 환경에서 Java 프로그래밍 언어와 SQL 서버를 사용하였다. 구현된 질의 수행 결과의 예를 몇 가지 보이면 다음과 같다.



(그림 8) 이력 테이블 검색 예

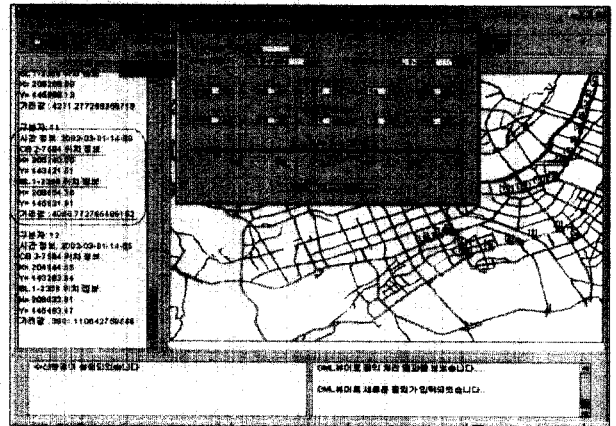
(그림 8)은 이력 릴레이션에 저장된 위치정보를 검색한 결과를 사용자 인터페이스를 통해서 출력한 화면이다. 이력 테이블에 저장된 검색뿐만 아니라 동일한 질의 매뉴를 통해서 객체의 일반 속성 정보 및 현재 정보 테이블도 검색이 가능하다.



(그림 9) 특정 시간의 이동 경로 검색 예

(그림 9)는 차량번호가 'SL 1-2358'인 객체가 '2002년 3월 1일 08시 00분~2002년 3월 1일 09시 00분' 까지 이동한 궤적 (trajectory) 즉, 이동 경로를 검색한 결과 화면이다. 출력 화면의 왼쪽 텍스트 창에는 데이터베이스에 저장된 차량의 위치 좌표를 보이고, 오른쪽의 지도 화면에는 이동 지점을 표시하고 있다.

(그림 10)은 차량번호가 'CB 2-7584'인 객체와 'SL 1-2358'



(그림 10) 이동 차량의 거리 검색 예

인 객체가 '2002년 3월 1일 14시 00분~2002년 3월 1일 15시 00분' 까지 이동한 경로와 두 객체간의 거리를 검색한 결과 화면이다. 화면의 왼쪽 텍스트 창에는 동일한 유효 시점에서 두 이동 객체의 위치 좌표 및 거리 계산 결과가 나타나며, 오른쪽 화면에는 두 객체의 이동 경로를 보이고 있다.

6. 결 론

차량 위치 추적을 위해 지금까지 연구된 이동 객체 관리 시스템들은 하나의 데이터베이스에서 과거 및 미래의 위치 정보를 동시에 관리하지 못한다. 그리고 동일한 이력 위치 정보를 이용하여 데이터베이스에 저장되지 않은 이동 객체의 불확실한 과거 및 미래의 위치 추정 방법을 제시하지 못하고 있다. 따라서 이 논문에서는 이와 같은 단점을 보완할 수 있는 이동 객체 관리 시스템을 제안하였다. 이를 위해 먼저, 이동 객체 관리 시스템에 적용될 차량 위치 정보를 모델링 하였다. 그리고 이동 객체 연산 처리기, 불확실한 위치 처리기, 실시간 차량 위치정보 수신 및 저장기, 이동 객체 데이터베이스로 구성된 이동 객체 관리 시스템 구성 및 처리 알고리즘을 제시하였다. 아울러 제안 시스템의 실행 모델 및 가상 시나리오를 적용한 구현 결과를 보였다.

제안 시스템은 다음과 같은 특징을 가진다. 첫째, 기존에 개별적으로 관리되던 이동 객체의 과거 및 미래의 위치정보를 하나의 데이터베이스를 이용하여 동시에 관리한다. 둘째, 데이터베이스에 저장되지 않은 과거의 위치 정보를 추정한다. 셋째, 차량 위치 탐지기의 통신 및 기타 장애로 인해 위치 정보를 제공받지 못할 경우 가까운 미래의 위치를 추정한다.

현재 이 논문에서 제안한 시스템은 우정 물류 관리를 위한 e-logistics 시스템 내의 이동 객체 관리 서버에 적용시킨 프로토타입으로 개발 중이다. 그런데 이동 객체의 위치 정보량은 시간에 따라 방대하게 증가되는 특성을 가진다. 따라서 향후에는 대용량 이동 객체 데이터 처리를 위한 인덱스 관리 방법에 대한 연구가 진행될 것이다.

참 고 문 헌

[1] 김종혁, "첨단 교통관리 시스템", 정보과학회지, 제16권 제6호, pp.5-13, 1998.

[2] 김형욱, 김성수, "물류 정보서비스를 위한 통신 매체와 이를 이용한 교통 물류관제 시스템의 전반적인 이해와 적용", 한국통신 Technical Memo, May, 1997.

[3] 안승범, "안전도향상을 위한 첨단 화물운송 시스템(CVO)의 서비스와 기술", 정보과학회지, 제16권 제6호, pp.30-35, 1998.

[4] R. Casey, "Automatic Vehicle Location Successful Transit Applications," A Cross-Cutting Study, in Intelligent Transportation Systems Electronic Document Library, Document No. 11487, U.S DOT Publication No.FHWA-JPO-99-022, Aug., 2000, <http://www.its.dot.gov/itsweb/welcome.htm>.

[5] M. Erwig, R. H. Güting, M. Schneider, and M. Vazirgiannis, "Abstract and Discrete Modeling of Spatio-Temporal Data Types," Chorochronos Technical Report, CH-98-14, 1998.

[6] M. Erwig, R. H. Güting, M. Schneider, and M. Vazirgiannis, "Spatio-Temporal Data Types : An Approach to Modeling and Querying Moving Objects in Databases", GeoInformatica Vol.3, No.3, pp.269-296, 1999.

[7] R. H. Güting, and et. al, "A Foundation for Representing and Querying Moving Objects," ACM Transactions on Database Systems, Vol.25, No.1, pp.1-42, 2000.

[8] L. Forlizzi, R. H. Güting, E. Nardelli, and M. Schneider, "A Data Model and Data Structures for Moving Objects Databases," Proc. of the ACM SIGMOD Conference, pp.319-330, 2000.

[9] O. Wolfson, B. Xu, S. Chamberlain, and L. Jiang, "Moving Objects Databases : Issues and Solutions," Proc. of the 10th International Conference on Scientific and Statistical Database Management, SSDBM'98, Capri, Italy, pp.111-122, 1998.

[10] P. Sistla, O. Wolfson, S. Chamberlain, and S. Dao, "Modeling and Querying Moving Objects," Proc. of the 13th International Conference on Data Engineering, ICDE'97, Birmingham, UK, Apr., 1997.

[11] P. Sistla, O. Wolfson, S. Chamberlain, and S. Dao, "Querying the Uncertain Position of Moving Objects," Springer Verlag Lecture Notes in Computer Science No.1399, pp. 310-337, 1998.

[12] O. Wolfson, P. Sistla, B. Xu, J. Zhou, S. Chamberlain, N. Rishe, and Y. Yesha, "Tracking Moving Objects Using Database Technology in DOMINO," Proc. of NGITS'99, The 4th Workshop on Next Generation Information Technologies and Systems, Zikhron-Yaakov, Israel, pp.112-119, 1999.

[13] O. Wolfson, S. Chamberlain, S. Dao, L. Jiang, and G. Mendez, "Cost and Imprecision in Modeling the Position of Moving Objects," Proc. of the 14th International Conference on Data

Engineering, ICDE'98, Orlando, FL, Feb., 1998.

[14] S. Saltinis, C. S. Jensen, S. Leutenegger, and M. Lopez, "Indexing the Positions of Continuously Moving Objects," Proc. of the ACM SIGMOD Conference, pp.331-342, 2000.

[15] D. Pfoser, Y. Theodoridis, and C. S. Jensen, "Indexing Trajectories of Moving Point Objects," Chorochronos Technical Report, CH-99-3, Oct., 1999.

[16] D. Pfoser, C. S. Jensen, and Y. Theodoridis, "Novel Approaches in Query Processing for Moving Objects," Proc. of the VLDB Conference, pp.395-406, 2000.

[17] D. Pfoser and C. S. Jensen, "Capturing the Uncertainty of Moving Object Representations," Proc. of Advances in Spatial Databases, 6th International Symposium, SSD'99, pp.20-23, 1999.

[18] D. Pfoser and N. Tryfona, "Fuzziness and Uncertainty in Spatiotemporal Applications," Chorochronos Technical Report, CH-00-4, Feb., 2000.

[19] S. Grumbach, P. Rigaux, M. Scholl, and L. Segoufin, "The Design and Implementation of DEDALE," 1999.

[20] S. Grumbach, P. Rigaux, and L. Segoufin, "Spatio-Temporal Data Handling with Constraints," ACM GIS, 1998.

[21] I. B. Oh, Y. A. Ahn, E. J. Lee, K. H. Ryu, and H. G. Kim, "Prediction of Uncertain Moving Object Location," Proc. of International Conference on East-Asian Language Processing and Internet Information Technology, EALPIIT'02, Jan., pp.51-58, 2002.

[22] S. S. Park, Y. A. Ahn, and K. H. Ryu, "Moving Objects Spatiotemporal Reasoning Model for Battlefield Analysis," Proc. of Military, Government and Aerospace Simulation part of ASTC'01, pp.108-113, 2001.

[23] K. H. Ryu and Y. A. Ahn, "Application of Moving Objects and Spatiotemporal Reasoning," TimeCenter TR-58, 2001.

[24] M. H. Huh, Y. A. Ahn, and K. H. Ryu, "Moving Object Location Change Function using Cubic Spline Interpolation," Proc. of the 6th World Multiconference on Systemics, Cybernetics and Informatics, SCI'02, Vol.V, Computer Science I, pp.336-341, 2002.

안 윤 애



e-mail : yeahn@dblab.chungbuk.ac.kr

1993년 한남대학교 전자계산공학과 졸업
(학사)

1996년 충북대학교 대학원 전자계산학과
(이학석사)

1999년~현재 충북대학교 대학원 전자계산
학과 박사과정

관심분야 : 이동 객체 데이터베이스, 시공간 데이터베이스, 지리
정보 시스템, 지식기반 정보검색 시스템 등



김 동 호

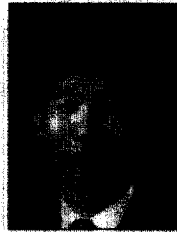
e-mail : kdh@etri.re.kr

1993년 충북대학교 전자계산학과(학사)

1995년 충북대학교 대학원 전자계산학과
(이학석사)

1999년 충북대학교 대학원 전자계산학과
(이학박사)

1999년~현재 한국전자통신연구원 우정기술연구센터 선임연구원
관심분야 : 시공간 데이터베이스, 이동체 기술, 지리정보시스템,
위치기반서비스, 물류응용정보시스템 등



류 근 호

e-mail : khryu@dblab.chungbuk.ac.kr

1976년 숭실대학교 전산학과 졸업(학사)

1980년 연세대학교 공학대학원 전산전공
(공학석사)

1988년 연세대학교 대학원 전산전공(공학
박사)

1976년~1986년 육군군수 지원사 전산실(ROTC 장교), 한국전자
통신 연구원(연구원), 한국방송통신대 전산학과(조교수)
1989년~1991년 Univ. of Arizona Research Staff(TempIS 연구
원, Temporal DB)
1986년~현재 충북대학교 전기전자 및 컴퓨터공학부 교수
관심분야 : 시간 데이터베이스, 시공간 데이터베이스, Temporal
GIS, 객체 및 지식기반 시스템, 지식기반 정보검색
시스템, 데이터마이닝, 데이터베이스 보안 및 Bio-
Informatics 등