효율적인 차량 이력 데이터 저장을 위한 유사 궤적 저장 기법

Similar Trajectory Store Scheme for Efficient Store of Vehicle Historical Data

한경복, 곽호영
제주대학교 통신컴퓨터공학부

Kyoung-Bok Han(oceanhan@cheju.ac.kr), Ho-Young Kwak(kwak@cheju.ac.kr)

요약

오늘날 무선 인터넷과 소형 이동 통신 기기 보급의 확산과 GPS 활용도의 급증으로 시간 변화에 따라 위치 정보가 연속적으로 변화하는 이동 객체의 연구가 활발히 이루어지고 있다. 그 중에서 차량 이동 객체에 대한 연구는 첨단 교통 정보 시스템, 차량 추적 시스템, 물류 수송 시스템에서 활용되고 있다. 이들 시스템들은 차량 이동에 대한 이력 데이터를 관리함으로서 과거의 차량 위치, 미래의 차량 위치 예측, 최적 경로, 최단 경로를 탐색 할 때 유용하게 사용되고 있다. 뿐만 아니라 물류 수송 계획과 차량 배차에도 차량 이력 데이터가 활용되고 있다. 이러한 차량 이력 데이터는 일정한 시간 간격을 갖고 저장되는데, 같은 패턴이 반복되는 차량 이력 데이터를 갖는 경우도 존재한다. 예를 들어, 특정 구간을 반복적으로 운행하는 차량일 경우에는 거의 유사한 경로로 운행을 한다. 이런 반복적인 운행경로를 일정 시간 간격 마다 차량 이력 데이터로 저장하면 많은 중복 데이터가 발생함으로써 저장 공간의 낭비를 유발한다. 따라서 본 논문에서는 이런 반복적인 운행경로를 갖는 차량에 대하여 이력 데이터를 효율적으로 저장할 수 있는 유사 궤적을 이용한 차량 이력 데이터 저장 기법을 제안하고자 한다.

■ 중심어 : | 유사 궤적 | 이동 객체 | 차량 이력 데이터 | 시공간 데이터베이스 | 시공간 이동 패턴 |

Abstract

Since wireless Internet services and small mobile communication devices come into wide use as well as the use of GPS is rapidly growing, researches on moving object, whose location information shifts sequently in accordance with time interval, are being carried out actively. Especially, the researches on vehicle moving object are applied to Advanced traveler information system, vehicle tracking system, and distribution transport system. These systems are very useful in searching previous positions, predicted future positions, the optimum course, and the shortest course of a vehicle by managing historical data of the vehicle movement. In addition, vehicle historical data are used for distribution transport plan and vehicle allocation. Vehicle historical data are stored at regular intervals, which can have a pattern. For example, a vehicle going repeatedly around a specific section follows a route very similar to another. If historical data of the vehicle with a repeated route course are stored at regular intervals, many redundant data occur, which result in much waste of storage. Therefore this thesis suggest a vehicle historical data store scheme for vehicles with a repeated route course using similar trajectory which efficiently store vehicle historical data.

■ keyword : | Similar Trajectory | Moving Object | Vehicle Historical Data | Spatio-temporal DataBases | Spatio-temporal Moving Pattern |

접수번호 : #051117-004
접수일자 : 2005년 11월 17일
심사완료일 : 2005년 11월 25일
교신저자 : 한경복, e-mail : oceanhan@cheju.ac.kr

I. 서 론

오늘날 무선인터넷과 휴대폰 및 PDA에 GPS 모듈이 내장된 이동 기기 보급의 확산으로 이동 객체에 관한 연구가 활발히 이루어지고 있다[1]. 그 중에서 시간에 따라 공간 특성이 빈번하게 변화하는 방대한 시공간 이동 데이터를 관리하기 위한 데이터베이스 기술과 위치 측위 기술이 발전함에 따라 위치기반 서비스[2], 위치기반 추천 시스템[3], 지능적 물류 관제 서비스[4], 차량 위치 추적 시스템[5] 등 새로운 시공간 응용 서비스 개발에 대한 관심과 연구가 증대되고 있다. 특히 차량 이동 객체에 대한 연구는 첨단 교통 정보 시스템, 차량 추적 시스템, 물류 수송 시스템에서 활용되고 있다. 이들 시스템들은 차량 이력 데이터를 관리함으로서 과거의 차량 위치 파악, 미래의 차량 위치 예측, 최적 경로, 최단 경로를 탐색할 때 유용하게 사용된다. 뿐만 아니라 물류 수송 계획과 차량 배차에도 활용되고 있다. 이러한 차량 이력 데이터는 일정한 시간 간격(Time Interval)을 갖고 위치 정보를 데이터베이스에 저장된다.

차량 이력 데이터는 차량의 운행 목적에 따라 운행 경로가 결정된다. 예를 들면, 택배 차량일 경우 출발지와 목적지가 지정되어 있어 특정 구간을 반복해서 운행을 한다. 또한 자가용 차량을 운전하는 운전자도 매일 아침 출발할 때 출발지와 목적지가 결정되고 퇴근할 때에도 출발지와 목적지가 결정된다. 다만 운행 중간에 어떠한 경로를 통하여 운행할지에 따라 약간의 경로가 달라질 수 있다. 그리고 택시를 운전하는 운전기사도 자신의 차고지에서 출발하여 택시 탑승을 기다리는 승객들이 많은 곳으로 이동한다. 택시 탑승을 기다리는 승객들이 있는 곳으로 이동할 때나, 승객이 택시에 탑승해서 승객의 목적지로 이동할 때에도 운전자가 특별히 자주 애용하는 경로들을 선택하여 승객을 목적지까지 이동한다. 이렇듯, 운전자들이 특별히 자주 애용하는 특정 경로들을 차량 이력 데이터에서 사용자 특징에 맞는 경로 패턴들을 시나리오로 만들고 이를 관리할 수 있는 차량 이력 데이터 저장기법에 대한 연구가 이루어지지 않고 있다.

지금까지 제안된 시스템에서의 차량 이력 데이터는 차량의 현재 이력 데이터들만 관리하고 있어, 차량의 과거 이력 데이터를 활용 할 수 없었다. 또한 차량 이력 데이터는 일정한 시간 간격을 통하여 입력되며 실시간으로 데이터베이스에 저장하기 때문에 대용량의 데이터베이스를 사용한다. 따라서 시스템 구축비용의 증가가 나타나고, 차량 검색에 오랜 시간이 소요된다. 그리고 기존의 차량 이력 데이터는 현재를 기준으로 짧은 시간 동안에 생성된 차량 이력 데이터만을 저장하고, 같은 경로를 반복해서 운행하는 차량 이력 데이터는 계속해서 반복된 데이터를 새로운 데이터로 저장하고 있어, 데이터 저장장소의 낭비를 유발하고 있다.

또한, 차량 이력 데이터들은 도시 및 도로 설계 시 반드시 참고해야 할 데이터이지만 실제 대용량의 차량 이력 데이터를 추출하고 사용하기란 매우 어려운 실정이다. 따라서 정규분포 및 가우스 분포 등을 이용해서 생성된 가상의 데이터를 사용함으로써 현실 세계를 반영함에 있어 부족한 점들이 많았다.

따라서 본 논문에서는 반복 운행하고 있는 차량 운행 경로를 체계적으로 관리할 수 있고, 차량 이력 데이터를 효율적으로 저장할 수 있는 차량 유사 궤적을 이용한 차량 이력 데이터 저장 기법을 제안하고자 한다. 이는 차량이라는 이동 객체의 특성과 속성을 반영하여 운전자가 자주 이용하는 차량 경로 패턴을 추출하여 이를 시나리오로 만들고, 만들어진 시나리오에서 유사 궤적으로 변환하여 저장하는 기법이다. 제안한 기법은 차량 이력 데이터 저장 공간을 절약할 수 있고, 특정 차량 검색 시간을 단축시킬 수 있을 뿐만 아니라, 실제 차량 이력 데이터를 관리함으로써 도시 및 도로 설계 시 도움을 주고자 한다.

본 논문의 구성은 1장 서론으로 앞서서 서술하였고, 2장은 관련 연구로 차량 이력 데이터와 관련된 이동 객체 관리 시스템에 대한 문제점을 제시한다. 3장에서는 차량 이동 객체의 위치 정보 모형의 정의와 차량 이동 객체의 특성과 시나리오 및 유사 궤적을 저장할 수 있는 데이터베이스 설계 및 차량 방향 정보의 표현 방법을 제시한다. 4장에서는 시나리오 및 유사 궤적 추출 알고리즘 제안하고 저장한다. 마지막으로 5장에서는 제안한 차량 이동 객체 시나리오와 유사 궤적 추출 알고리즘에 대한 문제점과 향후 연구 과제를 제시한다.

II. 관련 연구

차량 이동 객체는 시간에 따라 위치 정보가 연속적으로 변경되고 데이터베이스에 저장되는데 이러한 데이터를 차량 이력 데이터라고 한다. 현재 시간에 따라 객체의 공간 정보가 연속적으로 변하는 이동 객체에 대한 연구는 활발히 이루어지고 있지만 이동 객체의 이력을 관리하고 저장하는 기법에 대한 연구는 그리 많지 않다.

차량은 이동 객체의 한 부분이다. 따라서 지금까지 이루어진 이동 객체에 대한 연구에서 차량과 관련이 있는 연구들을 살펴보고자 한다.

이동 객체 관리를 위한 대표적인 응용 시스템 연구에는 DOMINO, CHOROCHRONOS, DEDALE, Battlefield Analysis가 있다. DOMINO[6-10]는 이동 객체의 수송에 대한 웹 기반 실시간 궤도 응용의 개발을 촉진하는 이동 객체 소프트웨어 도구로서, DBMS기술을 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(Trajectory)을 관리할 수 없는 단점을 가진다.

CHOROCHRONOS[11-15]는 시공간 데이터베이스의 특수한 형태인 이동 객체에 관한 연구가 집중적으로 수행되었고, 이동 객체의 데이터 모델링 및 인덱싱에 관한 연구 결과를 가장 많이 발표하였다. 특히 이 연구는 GPS기반의 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 이력 데이터 저장 기법이 제시되지 않고 있다.

DEDALE[16][17]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 대해 선형 제약사항 추상화를 제공하였다. 그러나 DEDALE 프로토타입은 데이터베이스에 시간의 변화에 따른 이동 객체의 이력 (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[18-20]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 하는 데 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 못하고, 동일한 데이터베이스에 저장된 이력 위치정보를 이용하여 이동 객체의 불확실한 과거 및 미래의 위치 추정방법을 제시하지 못하는 문제점을 가지고 있다. 또한 이력 데이터를 효율적으로 관리 저장하는 기법 연구들이 이루어지지 않았다.

III. 차량 위치 정보 모형

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

1. 차량 이동 객체 정의

본 논문에서는 이동 차량의 위치 정보 관리를 위해 이동점을 대상으로 하는 이동 객체만을 고려한 이동 차량

데이터를 다음과 같이 정의한다.

[정의 1] 차량 이동 객체

시간의 변화에 따라 객체의 위치 값만 변화되는 이동 객체를 말한다. 차량 이동 객체 VO는 시간 속성(Time Attribute), 공간 속성(Space Attribute), 일반 속성(General Attribute)을 가지며, 유일한 객체 식별을 위한 식별자를 포함하여 $VO = \langle VOID, S, T, A\rangle$로 표현된다.

[정의 2] 시간 속성

VO 시간 속성 $T_A = \langle vt_s, vt_e\rangle$로 구성되며 vt_s는 시작 시간, vt_e는 종료 시간을 나타낸다. 이 때 vt_s와 vt_e는 유효시간(Valid Time)의 집합 S_{VT}의 원소가 된다. 유효시간은 실세계에서 발생된 시간을 나타내며, $S_{VT} = \{t_0, t_1, t_2, \cdots, t_k, \cdots, t_{now}\}$이고, 각 원소들은 $t_0 < t_1 < t_2 \cdots < t_k < \cdots, \cdots < t_{now}$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 차량 이동 객체 데이터베이스는 동일한 유효시간의 주기(Granularity)를 가진다. 여기서 동일 유효시간 주기란 차량의 위치에 대한 유효한 시간정보를 표현할 때 발생 또는 저장하는 간격을 의미하며, 일반적으로 분 단위를 가장 많이 사용하지만 본 논문에서는 세밀한 데이터 구축과 적절한 데이터 분량을 가질 수 있도록 초 단위로 한다.

[정의 3] 공간 속성

VO의 공간 속성 $S_A = \langle x, y\rangle$, $x, y \in R$, R은 실수이다.

[정의 4] 이동 객체 데이터베이스

이동 객체 데이터베이스를 구성하는 이동 객체들의 집합은 $S_{VO} = \{VO_0, VO_1, \cdots, VO_n\}$이다. S_{VO}로 구성된 이동 객체 데이터베이스의 이력 집합은 $H_{VO} = \{H_{VO_0}, H_{VO_1}, \cdots, H_{VO_n}\}$이다. S_{VO}에 속하는 각각의 VO_i에 대한 모든 이력 집합은 $H_{VO_i} = \{vo_{i_0}, vo_{i_1}, \cdots, vo_{i_n}\}$이고,

$vo_{i_k} = \langle T_A(vo_{i_k}), S_A(vo_{i_k}), G_A(vo_{i_k})\rangle$이다.

[정의 4]에서 vo_{i_k}는 VO_i의 k번째 이력 정보를 의미하고, $T_A(vo_{i_k})$는 VO_i의 k번째 시간 속성, $S_A(vo_{i_k})$는 k번째 공간 속성, $G_A(vo_{i_k})$는 k번째 일반 속성을 의미한다.

2. 차량 이동 객체 연산자

이동 차량의 위치정보 관리를 위해 다음과 같은 이동 객체 연산자를 정의한다. 차량의 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_farthest를 [표 1]과 같이 정의한다.

표 1. 차량 위치 연산자

종류	입력 값	출력 값	기능
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 거리 계산
position_at	$G_A \times T_A$	S_A	임의의 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

[표 1]은 기존의 이동 객체 연구[21-24]에서 제시된 연산자들을 토대로 기본 연산자만을 정의한 것이다. 입력 및 출력 값에서 G_A는 차량 소유주, 운전자, 차종 등과 같은 이동 객체의 일반 속성을 의미한다. S_A는 공간 속성을 나타내고, T_A는 시간 속성인 유효시간을 의미한다. Real은 실수 값을 나타내며, { }로 묶인 속성들은

집합을 의미한다.

3. 데이터베이스 구조

제안한 저장기법에 사용되는 데이터베이스는 크게 4가지 테이블로 구성된다. 그 첫 번째 테이블은 시간 변화에 따른 차량 이동 객체의 ID와 차량 이동 좌표 값을 저장하고, 또한 차량 이동 좌표 값의 변화량을 저장하는 테이블이다. 테이블 구성은 [표 2]와 같다.

표 2. 차량 이동 좌표

필드명	Type	의미
mo_id	String	차량 ID
mo_time	Date	차량 시간
mo_x	double	차량 x좌표
mo_y	double	차량 y좌표
Sequence	String	x 변화값; y변화값

두 번째 테이블은 차량 객체를 운전하는 사람 즉, 사용자에 따라 차량을 운전하는 습관이 다르다. 예를 들면 차량을 출발 시킬 때 급가속을 하는 사람이 있는 반면 서서히 출발하는 사람도 있다. 따라서 가족 내에서도 주로 차량을 운전하는 사람 있고, 가끔 차량을 운행하는 사람이 있을 것이다. 이런 운전 패턴이 다른 사람들이 한 차량을 운전을 했을 경우 서로 판이한 운전 패턴을 갖게 된다. 운전자 속성 테이블은 이러한 운전자 속성 정보를 저장함으로써 주 운전자와 보조운전자의 패턴을 따로 저장하여 이동 패턴을 추출할 때 사용되는 테이블이다. 테이블 구성은 [표 3]과 같이 구성이 된다.

표 3. 운전자 속성 테이블

필드명	Type	의미
mo_user	String	주 운전자 이름
mo_user_no	String	주 운전자 주민번호
mo_id	String	차량ID
user_age	number	주 운전자 나이
user_sex	String	주 운전자 성별
user_Offic_addr	String	주 운전자 직장 주소
user_Home_addr	String	주 운전자 집 주소
mo_differ_user	String	보조 운전자

세 번째 테이블은 차량에 대한 속성 정보를 저장하는 테이블이다. 차량 속성 정보는 운행 목적에 따라 시나리오 추출에 중요한 요인이 될 수 있다. 예를 들면 출·퇴근 목적으로 차량을 운행하는 차량이라면 시나리오는 크게 출근 시나리오와 퇴근 시나리오만 생각할 수 있다. 하지만 영업용 목적으로 차량을 운행한다면 차량의 시나리오는 여러 가지 시나리오가 나올 수 있다. 따라서 차량 속성 정보를 저장할 수 있는 테이블이 필요하다. 차량 속성 정보는 [표 4]와 같이 구성된다.

표 4. 차량 속성 테이블

필드명	Type	의미
mo_id	String	차량 ID
mo_user	String	운전자 이름
mo_user_no	String	운전자 주민번호
mo_vol	String	차량의 용량
mo_op_purpose	String	차량 운행 목적
mo_kind	String	차종

[표 5]는 각 차량에 따른 시나리오를 저장하는 테이블이다. 시나리오의 시작과 끝을 지정해 줌으로써 시나리오를 관리 할 수 있도록 한다.

표 5. 차량 이동 패턴에 따른 시나리오 테이블 구조

필드명	Type	의미
mo_id	double	차량 ID
Name	String	차량 이름
VTS	String	시나리오 시작 시간
VTE	String	시나리오 끝 시간
SX	double	시나리오 시작 x좌표
SY	double	시나리오 시작 y좌표
EX	double	시나리오 끝 x 좌표
EY	double	시나리오 끝 y 좌표

표 6. 시나리오 유사 궤적 테이블

필드명	Type	의미
S_ID	String	시나리오 ID
S_x	String	x좌표
S-y	String	y좌표
S_SAn	number	경로 Sequence 각도
S_SRan	number	경로 Sequence 간격

[표 6]은 시나리오에 따른 유사 궤적 패턴을 저장, 관리하는 테이블이다. 유사 궤적은 5장에서 제안한 유사 궤적 패턴을 이용하여 궤적 패턴을 저장한다.

표 7. 시나리오 유사 궤적 버전

필드명	Type	의미
S_verID	String	시나리오 유사 궤적 버전ID
S_Date	Date	시나리오 유사 궤적 생성 일자
mo_id	String	차량 ID
S_frq	number	시나리오 유사 궤적 발생 빈도

[표 7]은 시나리오 유사 궤적 버전 관리 테이블이다. 유사 궤적을 저장함으로써, 개인의 유사 궤적 발생 빈도를 측정하여 [표 2]에서 제시된 차량 이동 좌표 제거하기 위함이다. 이는 차량 이동 좌표를 계속해서 저장한다면 많은 데이터베이스 용량을 소비할 것이다. 거의 같은 시나리오의 차량 이동 좌표를 계속해서 저장하는 것은 중복된 데이터를 저장하는 결과와 같다. 따라서 차량 이동 좌표를 유사 궤적으로 저장하여 버전으로 관리하면 데이터베이스 저장장소를 절약할 수 있고 또한 검색 시간을 단축할 수 있다.

4. 차량 방향 정보의 표현

방향 정보의 변경 기본적으로 시공간 이동 객체의 이동에 따라 이동 방향을 나타내는 각 θ가 변경되고, 이 각 θ가 주어진 오차의 범위를 벗어날 경우 행위 정보의 변경으로 보고 명시적 갱신 연산을 수행하는 것이다. 점 P는 시공간 이동 객체의 이동 위치를 나타내며, O는 기준선을, θ는 이동 방향을 나타낸다. O의 X좌표는 P의 X좌표이며 O의 Y좌표는 기준점의 Y좌표이다. 각 θ는 명시적 갱신이 수행되는 시점에 설정된 기준 각이다. 또한 θ_n은 기준선 O(또는 O_n)와 이동 객체의 위치 P_n 사이의 각을 나타내며 θ_1에서 θ_4로 변한다고 가정한다. 두 방법 모두 동쪽을 0° 로 한다. 방향 정보의 변화를 판단하는 방법으로 본 논문에서는 두 가지 방법을 사용한다.

첫 번째 방법은 n번째 이동 정보의 입력에서 시공간 이동 객체의 이동 방향을 나타내는 각 θ_n은 n번째 이동 정보의 위치와 $n-1$번째 이동 정보의 위치 그리고 기준선 O_{n-1}이 이루는 각을 구하여 이를 행위 정보에 기록된 방향 정보와 비교하는 것에 의해 이동 방향의 변경 여부를 판단한다. 이 방법에서는 기준선이 동적으로 구해지므로 각각의 이동 정보 입력마다 새로운 기준선 O_{n-1}의 위치를 계산하는 과정이 필요하다. n번째 이동 정보의 이동 방향을 나타내는 각 θ_n은 식(1)로 표현할 수 있다.

$$\theta_n = \angle P_n P_{n-1} O_{n-1} \text{(단, } n \geq 1) \quad (1)$$

다음의 [그림 1]에서 이동 정보 P1이 입력될 때 행위 정보에 기록된 방향 정보인 θ와 현재 입력하는 이동 정보의 이동 방향을 나타내는 θ_1의 차이를 계산하여 주어진 오차의 범위를 넘으면 행위 정보를 변경한다. P_2 역시 θ와 θ_2의 차이로 행위 정보 변경 여부를 판단하며 이후는 모두 동일하다.

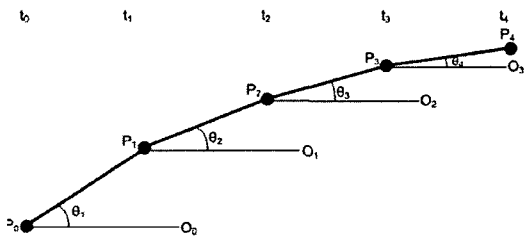


그림 1. 동적 기준선

만일 P_4의 입력 시점에 행위 정보가 변경되었다고 가정하면, 행위 정보의 위치 정보에는 P_4, 방향 정보에는 θ_4가 설정된다.

두 번째 방법은 행위 정보의 명시적 수정 연산 수행 시 설정된 시작 위치와 입력된 이동 정보의 위치사이의 각을 기준으로 이동 방향의 변화를 판단한다. 이 방법에서 기준선 O는 항상 고정적이다. n번째 이동 정보의 이동 방향을 나타내는 각 θ_n은 식(2)로 표현할 수 있다.

$$\theta_n = \angle P_n P_o O \quad (\text{단}, n \geq 1) \qquad (2)$$

다음의 [그림 2]에서 점선은 16방위 중의 한 부분으로 동일한 방향으로 처리되는 각의 범위, 즉, 오차의 범위이다. θ_n은 P_n의 이동 방향을 나타낸다.

[그림 2]에서는 이동 방향을 나타내는 각 θ_4가 오차의 범위를 벗어나는 시점인 t_4의 입력에서 행위 정보의 변경이 발생한다. 이때 행위 정보의 위치 정보에는 P_4의 우치가 방향 정보에는 P_4가 위치하는 영역을 나타내는 방위의 대표 값이 설정된다.

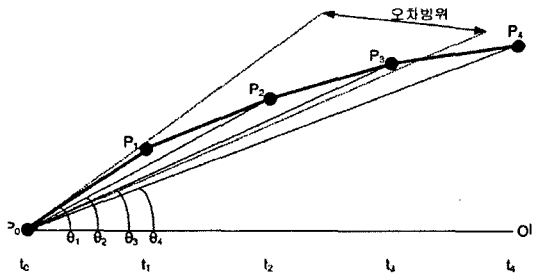


그림 2. 고정 기준선

두 가지 방법 모두 각각의 이동 정보 입력에서 시공간 이동 객체의 이동 방향을 나타내는 θ_n을 구해야 한다. 그러나 θ_n을 이루는 세 점의 특성은 두 방법에 차이가 있다. θ_n은 $\angle P_1P_2P_3$라고 하면, 첫 번째 방법은 P_1, P_2, P_3 모두가 계속적으로 변경된다. 즉 이들 세 점을 구하기 위한 추가적인 연산이 필요하다. 반면에 두 번째 방법은 P_2와 P_3는 고정적이며 P_1만을 구하면 되므로 첫 번째 방법보다 추가적인 연산이 적다. 두 방법에서 행위 정보로 기록되는 방향 정보의 값을 살펴보면 첫 번째 방법에서는 특정 크기의 θ을 가지며 오차의 크기는 응용 프로그램에서 정의되고, 두 번째 방법은 해당 방위별 대표 값이 저장된다. 두 방법은 응용의 특성에 따라 선택 가능하다.

Ⅳ. 시나리오 및 유사 궤적 추출

시나리오는 NAVIUS사의 NSA-U3 모델의 GPS 수신기와 (주)이스트소프트사의 ALMap SDK 지도를 이용하였다. 또한 데이터 저장은 Oralce사의 Oracle 9i를 이용하였으며 알고리즘 및 데이터 처리 방법은 Java 프로그래밍 언어로 구현하였다.

매일 아침 8시에 차량을 이용하여 그림 3과 같은 경로를 통하여 출근하는 운전자가 있다. 출발지와 목적지는 거의 대부분 같다. 이 차량의 경로는 매우 유사한 위도, 경도 좌표 값을 갖는다. 유사 경로 상의 차량의 속도 변화, 인공위성의 좌표 오차 범위 때문에 동일 위치 좌표 값을 갖는 확률은 희박하다. 따라서 차량의 이동 패턴을 추출하고 검색하고자 할 때는 오차 범위를 지정하여 검색하는 것이 좋다.

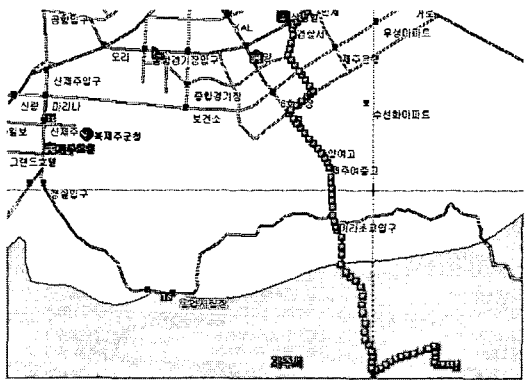


그림 3. 차량 이동 경로

[그림 3]과 같은 차량 이동 경로를 위한 시나리오는 크게 두 가지 제약 조건을 갖는다.

첫 번째가 거리 제약 조건(Distance Constraint) min_dist이다. 거리 제약 조건은 차량이 이동하였을 때 최소 이동한 거리를 말하며, 본 논문에서는 dist()함수를 이용하여 두 이동간의 공간상의 Eucledian거리를 구하였는데 도로 여건과 건물의 여건상 가장 효율적인 거리인 5m로 지정하였다. 즉, 차량이 이동한 거리가 5m 이내이면 이동하지 않고 정지하고 있음을 말한다.

두 번째 최대 시간 간격(Max Time Gap)조건 이다. 최대 시간 간격은 정지한것인지, 정차했는지를 결정하고 또한 시나리오 끝을 나타낸다. 최대 시간 간격을 벗어났을 경우 주차한 것으로 판명되도록 하였고 또한 시나리오가 끝나고 다른 시나리오가 생성되도록 하였다.

본 논문에서는 출·퇴근 차량을 이용하였기 때문에 5분으로 잡았다. 신호대기 시간은 이보다 적은 시간을 요하기 때문이다.

앞에서 제시된 제약 조건을 만족하는 차량 이동 객체의 시퀀스 데이터 즉, 경로 이력 데이터는 데이터베이스에 저장된다.

이렇게 저장된 시나리오 데이터는 [그림 4]에 제시된 각도 추출 알고리즘을 통하여 각 차량 이동 행위 정보를 추출한다.

```
public boolean AngleAbstraction(int ID, int x, int y){
   DBmanager dm = new DBmanager();
   /* Base Location */
  int baseX = dm.getBaseX(ID);
  int baseY = dm.getBaseX(ID);

  /* Current Location */
  int currentX = x;
  int currentY = y;

 /* Get Length of current to base */
int line1 = this.getLength(baseX, baseY, currentX, currentY);

 /Get Length of base line */
 int line2 = this.getLength(baseX, baseY, currentX, baseY);

 /*Get eaceh angles */
 int currentAlgle = this.GetAngle(line1, line);
 int baseAlgle = dm.getBaseAngle(ID);
 int diff = abs(currentAlgle - baseAlgle)
 /* Compare */
 if(diff ) dm.getBaseOrError(i(D))
     return true
 else
   return False
}
```

그림 4. 차량 이동 행위 정보 추출 알고리즘

[그림 5]는 [그림 4]에서 제시된 알고리즘에 의하여 추출된 행위 정보를 그림으로 표시한 것이다.

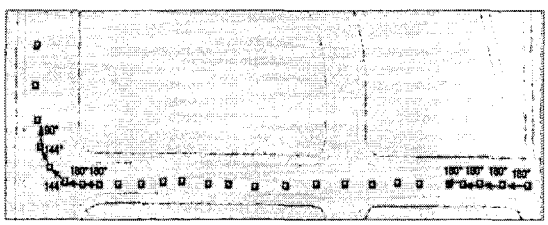


그림 5. 차량 이동 객체의 각도 추출 결과

[그림 5]에서 추출된 두 좌표 간 행위 각은 유사 궤적을 추출하기 위해서 또 다시 정제된다. 여기서는 두 각이 동일 각인 경우와 두 각의 차가 크게 벗어나지 않는 한 동일 각으로 지정하여 동일각의 시작과 끝 두 좌표 값만 데이터베이스에 저장하도록 하였고, 또한 선분을 그리도록 하였다

[그림 6]은 위에서 제시된 정제 작업을 통하여 그려진 차량 궤적이다.

[그림 6]에서 추출된 차량 궤적 데이터를 좀 더 효율적인 검색과 저장장소 감소를 위해서는 유사 궤적으로 만들어서 저장해야 한다.

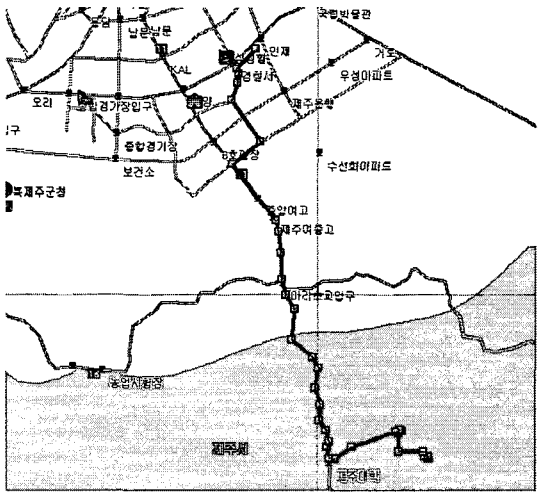


그림 6. 차량 궤적 추출

유사 궤적은 차량 이동 객체의 궤적을 이루는 각각의 움직임들을 단일 속성인 실제 각도(0~360)로 표현하고, 사용자 궤적 $Q = (q_1, q_2, \cdots, q_M)$을 추출한다.

차량 이동 객체의 궤적 T를 구성하는 움직임 요소(t[i])에 대해서 [그림 7]과 같이 방향(Direction) 정보와 거리(Distance) 정보를 이용하여 모델링한다.

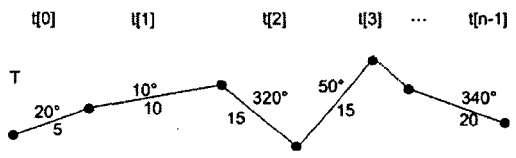


그림 7. 차량 이동 객체의 궤적 T

주어진 이동 객체의 궤적 정보에서 움직임 정보를 위해 방향 정보만을 추출하여 방향 궤적 T_{ang}을 구성한다.

$$T_{ang}=\{a_i | i=0,\cdots,n-1\},\ n=|T_{ang}| \quad (3)$$

여기서, a_i는 방향 궤적 T_{ang}를 구성하는 i번째 움직임 요소의 방향 정보를 나타내며 실제 각도[0, 360]를 이용하여 표현한다. n(=| T_{ang}|)은 궤적을 구성하는 움직임의 수를 의미한다. 궤적을 구성하는 임의의 움직임 요소 a_i와 a_{i-1}에 대해서 ADF(Angle Difference Function)함수를 이용하여 두 움직임 요소 사이의 각도의 차(Angle Difference : AD)를 계산한다.

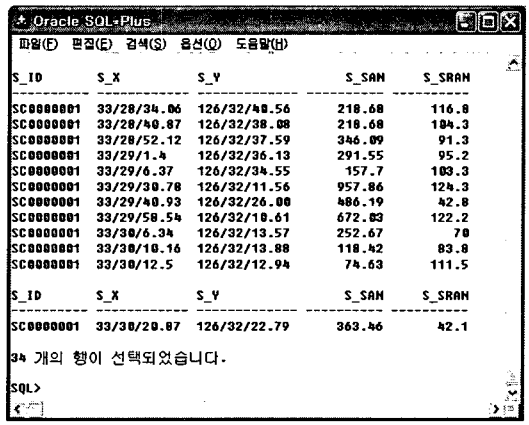
$$AD_i=ADF(a_i, a_{i+1}),\ i=0,\cdots,n-2 \quad (4)$$

```
int ADF(int a1, int a2)
{
    int Ang_Diff;
    Ang_Diff = 180°  + (a2 - a1);
    if(Ang_Diff 〉 360° ) Ang_Diff = Ang_Diff - 360° ;
    else if(Ang_Diff 〈 0) Ang_Diff = Ang_Diff + 360° ;
    return Ang_Diff;
}
```

위에서 구해진 각도 값들은 전절에서 제시된 데이터베이스에 저장된다. [그림 8]은 최종적으로 구해진 유사 궤적 정보를 저장한 그림이다.

지금까지 대량의 차량 이력 데이터에서 시나리오를 추출하고 추출된 차량 시나리오를 유사 궤적 이력 데이터로 저장하는 방법을 제안하였다.

제안한 차량 이력 데이터를 유사 궤적 이력 데이터로 저장하게 되면 데이터 저장장소가 감소한다. 시간간격(Time interval)이 1초인 출근 시나리오 데이터 개수는 출근 시간이 20분일 때 약 1100개정도의 레코드 데이터가 발생하고 저장된다. 하지만 유사 궤적 이력 데이터로 저장하면 최대 50개 이내로 저장장소가 감소하는 것을 보였다. 저장장소의 감소는 차량의 이동 경로에 따라 달라질 수 있는데 고속도로나 고속화 도로에서는 직선도로가 많기 때문에 유사 궤적 추출 포인트가 줄어든다. 하지만 방향 전환이 많은 도심지 도로에서는 유사 궤적 추출 포인트가 많아지는 경향뿐만 아니라 감소하는 경향도 보인다. 그것은 도심지 경로는 블록 단위로 도로가 설계되어 있는 곳에서는 의외로 직선 형태의 경로가 발생하는 경우도 있다. 즉 경로에 따라 유사 궤적 추출 포인트는 어떤 경로를 선택하는지에 따라 많은 편차를 보이고 있다.


Oracle SQL*Plus

파일(F) 편집(E) 검색(S) 옵션(O) 도움말(H)

S_ID	S_X	S_Y	S_SAN	S_SRAN
SC0000001	33/28/34.06	126/32/48.56	218.68	116.8
SC0000001	33/28/40.87	126/32/38.08	218.68	184.3
SC0000001	33/28/52.12	126/32/37.59	346.09	91.3
SC0000001	33/29/1.4	126/32/36.13	291.55	95.2
SC0000001	33/29/6.37	126/32/34.55	157.7	103.3
SC0000001	33/29/30.78	126/32/11.56	957.86	124.3
SC0000001	33/29/40.93	126/32/26.08	486.19	42.8
SC0000001	33/29/58.54	126/32/18.61	672.83	122.2
SC0000001	33/30/6.34	126/32/13.57	252.67	70
SC0000001	33/30/10.16	126/32/13.88	118.42	83.8
SC0000001	33/30/12.5	126/32/12.94	74.63	111.5

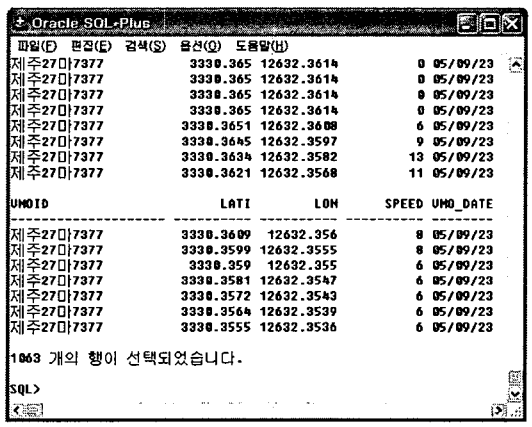
S_ID	S_X	S_Y	S_SAN	S_SRAN
SC0000001	33/30/29.87	126/32/22.79	363.46	42.1

34 개의 행이 선택되었습니다.

SQL>

그림 8. Oracle 데이터베이스에 저장된 유사 궤적 정보

[그림 9]는 시나리오에 대한 실제 차량 이력 데이터 레코드를 보여 주고 있다.


Oracle SQL*Plus

파일(F) 편집(E) 검색(S) 옵션(O) 도움말(H)

제주27마7377	3330.365	12632.3614	0	05/09/23
제주27마7377	3330.365	12632.3614	0	05/09/23
제주27마7377	3330.365	12632.3614	0	05/09/23
제주27마7377	3330.365	12632.3614	0	05/09/23
제주27마7377	3330.3651	12632.3608	6	05/09/23
제주27마7377	3330.3645	12632.3597	9	05/09/23
제주27마7377	3330.3634	12632.3582	13	05/09/23
제주27마7377	3330.3621	12632.3568	11	05/09/23

VMOID	LATI	LON	SPEED	VMO_DATE
제주27마7377	3330.3609	12632.356	8	05/09/23
제주27마7377	3330.3599	12632.3555	8	05/09/23
제주27마7377	3330.359	12632.355	6	05/09/23
제주27마7377	3330.3581	12632.3547	6	05/09/23
제주27마7377	3330.3572	12632.3543	6	05/09/23
제주27마7377	3330.3564	12632.3539	6	05/09/23
제주27마7377	3330.3555	12632.3536	6	05/09/23

1063 개의 행이 선택되었습니다.

SQL>

그림 9. 차량 이력 데이터

기존에 연구된 차량 이력 데이터 추출 시간 간격은 5

분 간격으로 경로 좌표 값들을 추출하였는데 이 경우 시간 간격이 길어 획득되는 데이터는 감소할 것이며, 긴 시간 간격으로 인한 정확한 경로 궤적을 유추하기 힘들다. 대부분 경로는 선형 보간법을 이용하여 경로 궤적을 표시하며 이를 차량 이력 데이터로 사용하고 있어 정확한 차량 이력 데이터라고 말할 수 없다. 즉 그것은 시간 간격이 5분이면 도심지 도로에서는 다른 블록으로 방향 전환이 발생할 수 있기 때문에 5분 간격의 시간 간격으로는 정확 경로 궤적을 알 수 없다. 따라서 본 논문에서는 시간 간격을 1초로 설정하여 보다 정확한 경로 궤적을 추출하려고 노력하였다.

V. 결 론

본 논문에서는 차량 이동 객체가 이동하였을 경우 생성되는 차량 이동 경로 궤적에 대한 차량 이력 데이터를 효율적으로 저장 검색 할 수 있는 유사 궤적 저장 기법을 제안하였다. 제안한 유사 궤적 저장기법은 차량이 연속적으로 움직임을 보였을 때, 시간 간격을 두고 연속적으로 저장된 이동점 위치 정보들을 이산적인 이동점 위치 정보로 변환하고 그것을 다시 궤적 데이터를 구성하는 움직임 요소들의 수가 가변이라는 특징과 움직임 요소들의 순서 정보가 중요하다는 특징을 고려하여 이웃한 움직임 요소들 간의 각도의 차를 이용하여 위치 정보와 방향각 정보 그리고 거리 정보로 변환하여 이를 저장하는 기법이다.

이를 통해 차량 이력 데이터를 저장하는데 필요한 저장 공간을 절약할 수 있도록 하였고, 또한 차량의 전체 궤적 이력 데이터에서 차량 이동 객체의 시나리오를 추출하기위하여 다양한 제약 조건들을 제시하였으며, 차량 이동 객체의 정의와 이를 효과적으로 표현할 수 있는 차량 이력 데이터베이스를 설계하여 이를 저장할 수 있도록 하였다. 본 논문에서는 실제 차량에 GPS장비를 탑재하였고, 탑재된 GPS에서 추출된 전체 차량 이력 데이터에서 시나리오를 추출하였다. 또한 추출된 시나리오에서 유사 궤적을 추출하여 비교 검색한 결과 시나리오로 추출된 차량 이력 데이터보다 유사 궤적 이력 데이터로 저장했을 때가 데이터 저장장소가 절약됨을 알 수 있었다.

향후 연구과제로는 다량의 차량 이력 데이터에서 다양한 시나리오를 추출할 수 있는 기법과 유사 궤적을 추출할 수 있는 다양한 기법들이 연구되어야 할 것이며, 또한 운전자의 다양한 행동 패턴에 의해 결정되는 차량 경로 패턴을 추출할 수 있는 연구가 병행되어야 한다. 그리고 유사 궤적을 추출하였을 때 차량 유사 궤적 검색을 위하여 다양한 유사성 측정 기법도 연구가 되어야 할 것으로 사료된다.

참 고 문 헌

[1] 이준욱, 남광우, "이동 객체 위치 일반화를 이용한 시공간 이동 패턴 탐사", 정보처리학회논문지 D, 제10-D권, 제7호, p.1103, 2003.

[2] 안병익, "LBS기술동향과 전망 LBS 구조 및 구성", 한국지리정보, 10월호, pp.52-56, 2001.

[3] 안윤애, 김동호, 류근호, "차량위치 추적을 위한 이동 객체 관리 시스템의 설계, 정보처리학회논문지D, 제9-D권, 제5호, 2002.

[4] 이금우, "위치기반 서비스를 위한 개인화된 추천 시스템", 이학 석사학위논문, 충북대학교, 2002.

[5] 김진석, 김동호, 류근호, "차량 궤적 추적을 위한 불확실성 처리기 구현, 정보처리학회논문지D, 제11-D권, 제5호, p.1167, 2004.

[6] 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.

[7] 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.

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

[9] O. Wolfson, P. Sistla, B. Xu, J. Zhou, S. Chamberlain, N. Rishe, and Y. Yesha, "Tracking Moving Object 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.

[10] 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.

[11] S. Saltenis, 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.

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

[13] 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.

[14] 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

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

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

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

[18] 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.

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

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

[21] X. Chen and I. Petrounias, "A framework for temporal data mining," In Proc of the 9th International Conference on Database and Expert System Applications, 1998.

[22] J. F. Roddick and M. Spiliopoulou, "Temporal data mining : survey and issues," Research Report ACRC-99-007, University of South Australia, 1999.

[23] K. Koperski and J. Han, "Discovering of Spatial Association Rules in Geographic Information Databases," In Proc. of the 4th International Symposium on Large Spatial Databases, 1995.

[24] K. Koperski, J. Han and J. Adhikary, "Mining knowledge in geographical data," to appear in Communications of the ACM, 1998.

저 자 소 개

한 경 복(Kyoung-Bok Han) 정회원

- 1994년 2월 : 제주대학교 해양학과(이학사)
- 2000년 2월 : 제주대학교 정보공학과(공학석사)
- 2001년 3월~현재 : 제주대학교 박사과정
- 2001년 3월~현재 : 제주대학교 시간강사

<관심분야> : 객체지향 프로그래밍, 프로그래밍 언어론, 모바일 프로그래밍, GIS

곽 호 영(Ho-Young Kwak) 정회원

- 1983년 2월 : 홍익대학교 전자계산학과(이학사)
- 1985년 2월 : 홍익대학교 전자계산학과(이학석사)
- 1991년 2월 : 홍익대학교 전자계산학과(이학박사)
- 1990년~현재 : 제주대학교 통신컴퓨터공학부 교수

<관심분야> : 객체지향 프로그래밍, 프로그래밍 언어론, 모바일 프로그래밍, GIS