

# 이동 시퀀스 트리를 이용한 효율적인 시공간 이동 패턴 탐사 기법

이 연 식<sup>†</sup> · 고 현<sup>††</sup>

## 요 약

최근 이동 객체의 동적인 위치나 이동성에 기반하여 여러 분야에 적용가능한 위치 기반 서비스를 개발하고자 다양한 객체의 이동 패턴들로부터 유용한 패턴을 추출하기 위한 패턴 탐사 기법에 대한 연구가 활발히 진행되고 있다. 이동 패턴 탐사는 특성상 방대한 시공간 데이터의 분석 및 처리 방법에 따라 패턴 탐사의 성능이 좌우된다. 기존의 시공간 패턴 탐사 기법들[1-6,8-11] 중 일부는 이러한 문제를 해결하기 위한 방법을 제시하였으나, 패턴 탐사 수행 시간이나 패턴 탐사 시 사용되는 메모리량을 최소화하는데 있어 아직 부족한 실정이다. 이에 본 논문에서는 방대한 시공간 이동 데이터 집합으로부터 순차적이고 주기적인 빈발 이동 패턴을 효과적으로 추출하기 위한 새로운 시공간 이동 패턴 탐사 기법을 제안한다. 제안된 기법에서는 이동 객체의 이력 데이터로부터 해시 트리 기반의 이동 시퀀스 트리를 생성하여 빈발 이동 패턴을 탐사함으로써 탐사 수행 시간을 83%~93% 감소시키고, 시간 및 공간 속성을 가진 상세 수준의 이력 데이터들을 공간 및 시간 개념 계층을 이용하여 실세계의 의미있는 시간 및 공간영역으로 일반화함으로써 탐사 시 소요되는 메모리량을 감소시켜 보다 효과적인 패턴 탐사를 유도한다.

키워드 : 이동 객체, 시공간 패턴 탐사, 데이터 일반화, 이동 시퀀스 트리

## The Efficient Spatio-Temporal Moving Pattern Mining using Moving Sequence Tree

Yonsik Lee<sup>†</sup> · Hyun Ko<sup>††</sup>

## ABSTRACT

Recently, based on dynamic location or mobility of moving object, many researches on pattern mining methods actively progress to extract more available patterns from various moving patterns for development of location based services. The performance of moving pattern mining depend on how analyze and process the huge set of spatio-temporal data. Some of traditional spatio-temporal pattern mining methods[1-6,8-11] have proposed to solve these problem, but they did not solve properly to reduce mining execution time and minimize required memory space. Therefore, in this paper, we propose new spatio-temporal pattern mining method which extract the sequential and periodic frequent moving patterns efficiently from the huge set of spatio-temporal moving data. The proposed method reduces mining execution time of 83%~93% rate on frequent moving patterns mining using the moving sequence tree which generated from historical data of moving objects based on hash tree. And also, for minimizing the required memory space, it generalize the detained historical data including spatio-temporal attributes into the real world scope of space and time using spatio-temporal concept hierarchy.

Keywords : Moving Object, Spatio-Temporal Pattern Mining, Data Generalization, Moving Sequence Tree

## 1. 서 론

시공간 이동 패턴 탐사는 연속되는 시간상에서 발생하는 객체의 다양한 패턴들 중 독특한 이동 경향을 갖는 패턴이나 반복적이고 주기적인 빈발 패턴을 탐사하여 새로운 위치

기반 서비스를 개발하기 위한 기술이다. 현재까지 이러한 이동 패턴을 탐사하기 위해 많은 기법들이 제안되었으며, 대부분 주기 패턴[1,2]이나 순차 패턴[3-6,8-11]과 같은 빈발한 이동 패턴들의 탐사를 목적으로 하고 있다. 하지만 기존의 탐사 기법들은 패턴 탐사 시 후보 이동 시퀀스 생성 및 검증, 후보 시퀀스들의 지지도 계산을 위해 빈번하게 데이터베이스를 스캔함으로써 탐사 수행 시간이 많이 소요되는 단점이 있으며, 탐사 대상 데이터양이나 최소지지도 임계값에 의해 탐사 수행 시간이 좌우되는 특성이 있다. 또한, 많은 양의 데이터를 탐사 대상으로 하거나 탐사 과정 중 생성

※ 본 연구는 한국과학재단 특정기초연구(R01-2007-200989-0) 지원으로 수행되었음.

† 종신회원 : 군산대학교 컴퓨터정보공학과 교수

†† 정 회 원 : 한국항공우주연구원 프로젝트연구원

논문접수 : 2008년 5월 6일

수정일 : 1차 2009년 3월 3일, 2차 2009년 3월 27일

심사완료 : 2009년 3월 27일

되는 후보 시퀀스 집합이 클 경우 탐사 시 소요되는 메모리 양이 급격히 증가하는 문제가 있다. 이러한 문제들로 인해 일부 기법들은 잦은 데이터베이스 스캔에 따른 탐사 수행 시간을 감소시키기 위해서 패턴 트리를 생성하여 후보 이동 패턴에 대한 검증 및 지지도 계산을 수행하였고[1,2], 일부 기법들은 탐사 대상 데이터양에 따른 탐사 수행 시간 및 메모리 소요량을 감소시키기 위해서 데이터 일반화 방법을 사용하였다[1,5,8-11]. 하지만 데이터 일반화 방법에 있어 실제계의 시간 및 공간에 대한 고려없이 사용자 정의에 따라 물리적인 형태로 일반화를 수행하기 때문에 일반화된 데이터로부터 실제계의 공간 및 시간에 대한 의미를 얻는 것이 어렵고, 이력 데이터양에 따라 패턴 트리를 생성하고 유지하는 비용이 커지는 문제가 있다. 따라서 데이터베이스 접근을 최소화하여 탐사 수행 시간을 단축하거나 이동 객체의 데이터양, 최소지지도, 이동 패턴 트리 등과 같은 여러 요인으로 인한 메모리 소요량을 최소화하기 위한 새로운 기법이 필요하다. 또한 기존의 탐사 기법들은 시간 및 공간 한정자(qualifier)에 의해 제한된 시간 및 공간 영역에서의 모든 이동 패턴들 중 불특정한 빈발 이동 패턴만을 탐사하기 때문에 시간 및 공간 한정자 이외의 또 다른 시간 및 공간 제약 조건을 가진 시공간 이동 패턴을 탐사하는 문제에 대해서 능동적으로 대처하기 어렵다. 가령, 특정 지점들 간의 최적 이동 경로나 정해진 시간 내에서의 효율적인 스케줄링 경로 탐색과 같이 사용자에게 의해 지정된 시공간 범위 내에서 시간 및 공간 제약 조건을 갖는 빈발 패턴을 탐사하는 문제에는 적용하기 어렵다. 따라서 이러한 문제에 대한 근본적인 해결을 통해 보다 효율적인 패턴 탐사를 수행하는 새로운 시공간 이동 패턴 탐사 기법이 필요하다.

이에 본 논문에서는 방대한 시공간 이동 데이터 집합으로부터 순차적이고 주기적인 빈발 이동 패턴을 효율적으로 추출하기 위한 새로운 시공간 이동 패턴 탐사 기법인 STMPM\_MST(Spatio-Temporal Moving Pattern Mining\_Moving Sequence Tree) 기법을 제안한다. 제안된 기법에서는 이동 객체의 이력 데이터 집합으로부터 빈발 이동 패턴을 탐사하는데 있어 시간 및 공간 속성을 가진 상세 수준의 이력 데이터들을 공간 및 시간 개념 계층을 이용하여 의미 있는 시간 및 공간 영역 정보로 일반화하고, 해시트리 기반의 이동 시퀀스 트리(moving sequence tree)를 생성하여 빈발 패턴을 추출함으로써 탐사 수행 시간 및 소요 메모리 양

의 최소화를 통해 보다 효율적인 패턴 탐사가 수행될 수 있도록 지원한다.

본 논문의 구성은 2장에서 관련연구로서 기존의 이동 패턴 탐사 기법에 대해 고찰하고, 3장에서는 순차적이고 주기적인 빈발 이동 패턴을 탐사하는 문제에 대해 정의한다. 4장에서는 이동 시퀀스 트리의 생성을 통한 새로운 시공간 이동 패턴 탐사 방법을 제시하고, 5장에서는 실험을 통해 제안된 탐사 기법의 성능을 평가한다. 마지막 6장에서는 결론 및 향후 연구 과제를 제시한다.

## 2. 관련 연구

이동 객체의 시공간 이동 패턴 탐사는 연속되는 시간영역에서 이동하는 객체의 패턴들 중위치 변화에 따른 독특한 경향이나 반복적이고 공통적인 의미있는 패턴을 탐사하기 위한 기술이다. 현재까지 이러한 객체의 이동 패턴을 탐사하기 위한 많은 기법들이 제안되었다. 기존의 시공간 이동 패턴 탐사 기법들은 대부분 주기적인 이동 패턴[1,2]과 순차적인 이동 패턴[3-6,8-11]의 탐사를 목적으로 하고 있다. 하지만 기존의 탐사 방법들은 빈발 패턴 탐사 시 빈번하게 데이터베이스를 접근함으로써 탐사 수행 시간이 증가하는 문제와 탐사 대상 데이터의 양에 따라 탐사 시 소요되는 메모리 양이 증가하여 탐사 성능이 저하되는 문제가 있다. 다음 <표 1>은 패턴 탐사 방법 및 후보 시퀀스 검증 방법, 데이터베이스 접근 횟수, 데이터 일반화 지원여부 등 4가지 측면에서 각 탐사 기법들의 성능을 비교하였다.

<표 1>에서 STPMine1[2]와 UMP[6], Apriori-MSP[11], MP&GMP[5,8,9], MPMine[10]는 Apriori 계열의 알고리즘을 변형하여 사용하기 때문에 빈발 항목 탐사 시 후보 항목 집합의 크기가 클수록 지지도 계산 및 후보 항목들의 검증을 위해서 빈번하게 데이터베이스를 반복 스캔함으로써 탐사 수행 시간이 증가하는 문제가 있다. 하지만 MP&GMP와 MPMine의 경우 후보 항목의 생성 및 검증을 위해 해시트리를 통해 후보 시퀀스 집합을 저장하고 관리하기 때문에 다른 기법에 비해 탐사 수행 시간이 적게 소요되는 특징이 있다. 또한, STPMine2[2]와 STMPE[1,12]는 각각 개별적인 패턴 트리를 생성하여 패턴 탐사를 수행함으로써 데이터베이스에 대한 빈번한 접근을 감소시켜 탐사 수행 시간을 최소화하도록 하였으나, 유효시간 기준의 샘플링 방식을 통해

<표 1> 시공간 이동 패턴 탐사 기법들의 탐사 방법

기법	탐사방법	후보시퀀스 검증 방법	데이터베이스 접근 횟수	시간 및 공간 데이터일반화
STPMine1	Apriori 변형	지원안함	빈발	물리적(공간)
STPMine2	패턴트리	패턴트리	2회	물리적(시간/공간)
STMPE	패턴트리	패턴트리	2회	물리적(시간/공간)
UMP	Apriori 변형	지원안함	빈발	물리적(공간)
Apriori-MSP	Apriori 변형	지원안함	빈발	물리적(공간)
MP&GMP	Apriori 변형	해시트리	빈발	논리적(공간)
MPMine	Apriori 변형	해시트리	빈발	논리적(공간)

추출한 이력 데이터를 대상으로 탐사를 수행하기 때문에 일반적으로 위치 기준의 샘플링 방식을 통해 추출된 이력 데이터에 대해서는 변환 과정이 필요하다. 이 외에도 STPMine2의 경우 최소지지도와 데이터의 양에 따라 패턴 트리를 생성하고 유지하는 비용이 커지는 문제가 있다.

한편, 기존의 패턴 탐색 기법들은 탐사 대상 데이터에 대해 묵시적 또는 명시적으로 데이터 일반화를 통해 패턴 탐사를 수행하였다. <표 1>에서 STPMine1와 UMP, Apriori-MSP는 공간 속성에 대해서만 일반화를 수행하였고, STPMine2와 STMPE는 이동 객체의 시간과 공간 속성 모두에 대해 각각 일반화를 수행하였다. 하지만 이들 기법에서 적용한 일반화 방법은 실제계의 시간 및 공간 개념에 대한 고려없이 단순히 사용자 정의에 따라 물리적인 형태로 시간영역과 공간영역을 분할하여 각 영역 대표값으로 일반화를 수행하였다. 따라서 패턴 탐사를 통해 추출되는 지식을 활용하기 위해서는 지식 응용 분야별로 사용하는 개념적인 시간영역과 공간영역으로의 매핑이 필요하다. 또한, MP와 MPMine은 실제계의 공간지역에 대한 공간 개념 계층을 형성하여 이동 객체의 공간 속성에 대한 일반화만 수행하였기 때문에 시간 속성도 지식 응용 분야에서 적용되고 있는 시간 개념에 대한 시간영역으로의 일반화가 요구된다. 지금까지 고찰한 기존의 연구들에 대한 문제점을 해결하기 위하여 최소지지도와 이동 객체 수 등에 영향을 받지 않고 일정한 탐사 수행 시간을 유지하며 최소의 탐사 시간을 소요하는 새로운 시공간 패턴 탐사 기법의 개발이 필요하다.

### 3. 시공간 빈발 이동 패턴 탐사 정의

시공간 빈발 이동 패턴 탐사는 이동 객체의 위치가 시간에 따라 변화함으로써 나타날 수 있는 다양한 경향이나 패턴들 중 사용자의 이동 성향을 고려하여 위치 기반 정보 서비스 형태로 제공할 수 있는 의미있는 이동 패턴을 탐색하는 것이다. 일반적으로 이동 객체의 위치 데이터로부터 찾아낼 수 있는 지식의 예는 다음과 같다.

지식 1 : A 지점에 위치한 차량의 50%는 B지점을 지난다.
지식 2 : 변화가로 진입한 전체 차량의 25%는 A→B→C→D 지점순으로 이동한다.
지식 3 : 20대 운전자의 60%는 C→D→E→F 지점 순의 이동 패턴을 가진다.
지식 4 : 30대 운전자의 70%는 변화가에서 B→C→D 지점순으로 이동하는 경향이 있다
지식 5 : 오전 7시부터 9시 사이에 30%의 운전자가 A 지점에서 B지점으로 이동한다.

이러한 지식의 유형들은 이동 객체의 다양한 이동 패턴들 중 서로 연관적인 특성을 갖거나 주기적으로 혹은 반복적으

로 발생하는 빈발한 이동 패턴을 탐사함으로써 발견될 수 있다. 지식 1은 공간영역의 변화에 대한 연관성을 보여주며, 지식 2와 지식 3은 이동 객체의 순차적인 이동 패턴을 추출한 것이다. 지식 4와 지식 5는 각각 공간 제약 조건과 시간 제약 조건을 적용하여 이동 객체의 순차 패턴을 추출한 것이다. 예시로 보인 각 지식들은 기존에 이미 제안되었던 패턴 탐사 기법을 통해 추출할 수 있지만, 관련연구에서 고찰한 바와 같이 패턴 탐사 시 탐사 성능 측면에서 비효율적인 문제를 포함하고 있다. 따라서 이러한 문제를 해결하기 위한 새로운 시공간 빈발 이동 패턴 탐사 문제를 다음과 같이 정의한다.

#### [정의 1] 시공간 빈발 이동 패턴 탐사 정의

이동 객체 데이터베이스 MD, 사용자가 지정한 최소 지지도  $min\_sup$ , 공간 영역 간의 시간 간격 제약 조건  $max\_gap$ 이 주어졌을 때, 최소지지도를 만족하는 모든 빈발 이동 시퀀스를 탐색하는 것이다.

[정의 1]에서 최소지지도  $min\_sup$ 는 시퀀스  $s$ 가 빈발한 것으로 평가되기 위해 만족해야 하는 지지도의 하한값이다. 시간 간격 제약 조건은 이력 데이터 집합으로부터 트랜잭션 데이터 집합을 추출하기 위해 연속되는 객체의 이동 이력을 단위 이력, 즉 단위 이동 시퀀스로 분리하기 위한 제약 조건이다. 시간 간격 제약 조건  $max\_gap$ 은 객체의 연속적인 이동 이력 중 특정 영역에서 인접한 다른 영역으로의 이동 시간  $t_j - t_{j-1}$ 이 가질 수 있는 최대 시간 간격으로, 인접 영역으로의 이동 시간이  $max\_gap$  보다 크면 하나의 시퀀스를 두 개의 시퀀스로 분리한다. 즉, 단위 이동 시퀀스를 생성하기 위한 시간 간격 제약은  $t_j - t_{j-1} \leq max\_gap, (2 \leq j \leq k)$ 을 만족해야 한다.

### 4. 시공간 빈발 이동 패턴 탐사 기법

본 장에서는 앞서 제시한 시공간 빈발 이동 패턴 탐사에 대한 정의를 기반으로 이동 시퀀스 트리의 생성을 통해 효율적으로 빈발 패턴을 추출하는 STMPM\_MST 기법을 제시한다.

#### 4.1 시공간 빈발 이동 패턴 탐사 과정

시공간 빈발 이동 패턴 탐사 과정은 이동 객체의 이력 데이터를 변환하는 전처리 단계와 빈발한 이동 패턴을 탐사하는 패턴 탐사 단계로 구분된다. 전처리 단계에서는 상세 수준의 공간 및 시간 속성으로 구성된 이력 데이터를 패턴 탐사 수행에 적합한 형태로 변환하고, 빈발 이동 패턴 탐사 단계에서는 일반화된 시퀀스 집합으로부터 빈발 1-시퀀스 집합을 추출하고 이동 시퀀스 트리를 생성하여 최소지지도를 만족하는 빈발 패턴을 탐사한다.

4.1.1 데이터 전처리 단계

이동 객체의 이력 데이터에 대한 전처리 단계에서는 단위 이동 시퀀스 생성, 단위 이동 시퀀스의 공간 및 시간 속성 일반화, 일반화된 데이터의 요약 과정을 거쳐 패턴 탐사 수행에 적합한 형태로 데이터를 변환한다[13]. 다음 (그림 1)은 이동 객체의 이력 데이터를 전처리 과정을 통해 일반화-요약된 시퀀스 집합으로 추출하는 과정에 대한 예이다.

이동 시퀀스는 이동 객체의 이력 데이터를 각 객체별 시간 순차에 따라 나열한 순차 리스트로, 사건 사이의 시간 간격이 고려되지 않아 패턴 탐사를 위한 트랜잭션 데이터로 사용하기에는 문제가 있다. 따라서 시퀀스를 구성하는 위치들 간에 시간 간격에 대한 제약을 두어 이를 만족할 경우에만 의미있는 단위 이동 시퀀스로 생성한다. 시간 간격 제약 조건은 한 시퀀스 내에서의 연속적인 이동으로 객체가 특정 위치에서 인접한 다른 위치로 이동할 경우, 인접한 위치로의 이동시간  $t_j - t_{j-1}$ 이 최대 시간 간격  $max\_gap$ 을 초과하는지의 여부에 따라 초과 이전까지의 이동 시퀀스와 초과 이후의 이동 시퀀스로 분리한다[13]. (그림 1)에서  $max\_gap$ 을 1시간이라고 했을 때, (a)와 같이 두개의 이동시퀀스로 분리된다.

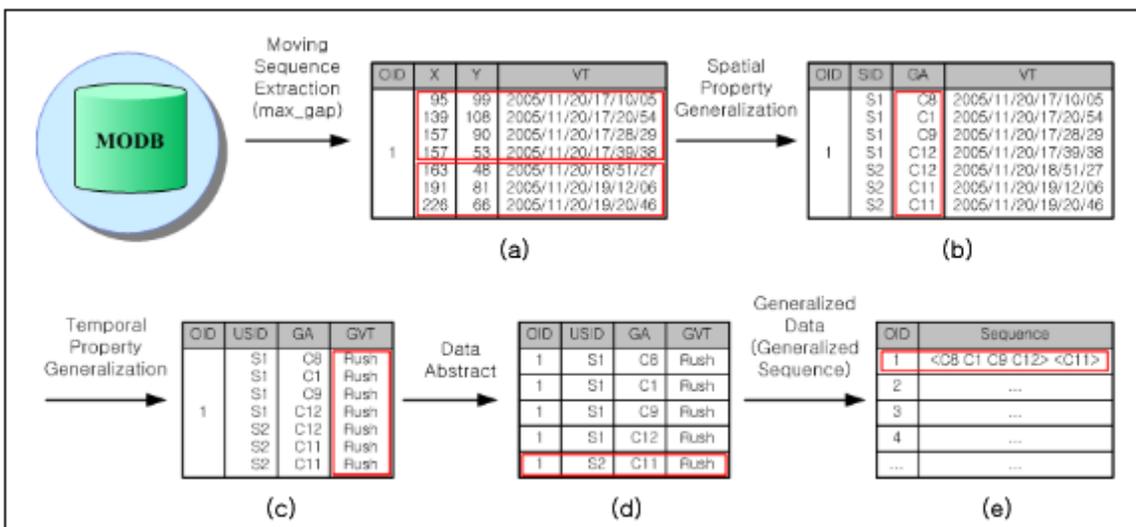
이동 객체의 공간 속성 일반화는 OGC(Open GIS Consortium)의 위상 관계 연산들 중 Contains 연산을 이용하여 일반화를 수행한다. Contains 연산은 공간 객체들 간의 포함 관계를 정의한 연산으로, 이동 객체의 위치값과 공간 영역 간의 포함 여부를 고려하여 객체의 위치 속성을 실제계의 공간 개념 계층에 따라 공간영역으로 일반화한다[14]. (그림 1)에서 (a)의 (X, Y) 공간속성값은 (b)의 공간영역으로 일반화된다. 또한, 시간 속성 일반화는 Allen[7]의 시간 간격 연산들 중 During 연산을 이용해 일반화를 수행한다. During 연산은 시간 영역들 간의 포함 관계를 정의한 연산으로, 객체가 공간상에서 이동할 때 이산적으로 샘플링된 유효시간과 시간영역 간의 관계 분석을 통해 시간 속성을

지식 응용 분야의 시간 개념 계층에 따라 시간영역으로 일반화한다[15]. (그림 1)에서 (b)의 VT 시간속성값은 (c)의 시간영역 Rush로 일반화된다. 이 때, Rush는 시간 개념 계층에서 Level 5 수준의 시간 영역이다. 마지막으로 일반화된 데이터의 요약은 연속된 시퀀스 항목들 중 공간 속성과 시간 속성이 동일한 경우 하나의 시퀀스 항목으로 요약한다. (그림 1)의 (c)에서 시퀀스ID인 SID가 S2이고, 일반화 공간영역(GA)이 C11, 일반화 시간영역(GVT)이 Rush인 두 이력 데이터는 (d)와 같이 하나의 이력으로 요약될 수 있다. 또한 일반화 및 요약된 이력 데이터 리스트는 (e)와 같이 시간영역이 Rush일 때, 두 개의 단위 시퀀스로 분리된다.

이러한 이동 객체의 공간 및 시간 속성 일반화는 (X, Y) 형태의 공간 속성값과 (년/월/일/시/분/초) 형태의 시간 속성값을 GA와 GVT와 같이 단순히 일반화된 공간/시간 구분 코드로 표현하기 때문에 데이터 크기를 축소시켜 탐사 과정에서의 소요 메모리량을 감소시킨다. 또한 데이터 요약도 중복 데이터를 제거하여 메모리 소요량을 감소시킨다.

4.1.2 빈발 이동 패턴 탐사 단계

빈발 이동 패턴이란 도로 네트워크에서 특정지점들 사이를 운행한 차량들의 다양한 이동 경로들 중 차량들이 최소 지지도 이상으로 공통적으로 이동한 경로를 의미한다. 보다 정확하게 정의하면, 공통적인 이동 경로를 운행한 차량들의 이동 패턴이라 할 수 있다. 빈발 이동 패턴 탐사는 이러한 차량의 빈번한 이동 패턴을 탐사하는 방법으로, 본 논문에서는 기존 Apriori 계열의 알고리즘과 같이 트랜잭션 데이터 베이스를 여러 번 스캔하여 후보 시퀀스 집합에 대한 지지도 계산을 통해 빈발 패턴을 탐사하는 대신 이동 시퀀스 트리를 구성하는 과정을 통해 빈발 패턴을 탐사한다. 이동 시퀀스 트리는 트랜잭션 데이터 집합의 각 객체별 시퀀스들을 헤시트리 기반의 시퀀스 트리로 구성하여 사용자가 지정한 최소지지도를 만족하는 빈발 패턴을 추출하기 위한 이동 패



(그림 1) 이동 객체의 이력 데이터 전처리 과정

턴 탐사 트리이다. 시퀀스 트리의 생성은 트랜잭션 데이터인 객체별 이동 시퀀스에 대한 모든 부분 시퀀스 집합을 추출한 후 각 부분 시퀀스를 구성하는 항목들을 노드로 생성하고, 각 항목들 간의 구성 순서에 따라 노드 간의 계층적인 연결 관계를 형성하여 트리를 구성한다. 이 때, 각각의 빈발 1-시퀀스 항목에 대해 하나의 트리가 생성되며, 각 빈발 1-시퀀스 항목이 루트노드가 된다. 가령, 빈발 1-시퀀스 항목 집합 {<N1> <N2> <N3> <N4> <N5> <N6>}이 있을 때, 이동 시퀀스 트리는 각 항목을 루트노드로 하여 총 6개의 트리를 생성할 수 있다. 생성된 각 시퀀스 트리는 개별적으로 하나의 노드 참조 테이블을 유지한다. 노드 참조 테이블은 키(key)와 값(value)의 쌍으로 구성되는 요소(element)를 저장하는 해시 테이블로서, 키값으로 노드 식별자인 NodeID를 가지며, 값을 저장하기 위한 버킷(bucket)은 트리의 깊이에 대응하는 이동 시퀀스들의 지지도(Support)와 부모노드 및 자식노드에 대한 참조자(ParentPtr, ChildPtr)를 저장한다. 다음 (그림 2)는 루트노드가 <N1>인 이동 시퀀스 트리와 트리를 구성하는 각 노드들에 대한 지지도 및 접근 정보를 저장하는 노드 참조 테이블에 대한 예이다.

빈발 이동 패턴 탐사 단계는 일반화-요약 시퀀스 집합으로부터 빈발 1-시퀀스 집합의 추출, 일반화-요약 시퀀스 집합의 트랜잭션 데이터로의 변환, 이동 시퀀스 트리의 생성, 빈발 이동 패턴 탐사 등의 과정을 통해 최소지지도를 만족하는 최대 빈발 패턴을 탐사하는 단계이다. 패턴 탐사 과정을 보이기 위해서 일반화 및 요약 과정을 통해 <표 2>의 이동 시퀀스 집합을 생성하였다고 가정한다. <표 2>의 시퀀스 집합은 도로 네트워크에서 운행하는 차량의 이력 데이터로부터 러시아워(Rush Hours) 시간대에 A5 공간영역에서 A11 공간영역으로 이동한 차량의 패턴들을 순차적으로 추출한 리스트 집합이다. 최대 빈발 패턴을 탐사하기 위한 각 과정은 다음과 같다.

**[단계 1] 빈발 1-시퀀스 집합 발견 및 트랜잭션 데이터 변환 단계**

빈발 패턴 탐사 과정 중 첫 단계는 빈발 1-시퀀스 집합을 추출하는 단계로, 일반화-요약 시퀀스 집합으로부터 최

소지지도 임계치  $min\_sup$  이상의 빈발 1-시퀀스 집합을 추출한다. <표 2>의 이동 시퀀스 집합으로부터 최소지지도 임계치가 2 이상인 빈발 1-시퀀스 집합을 추출하면 {<A5>, <A6>, <C1>, <C3>, <C8>, <C9>, <C11>, <C12>}와 같다.

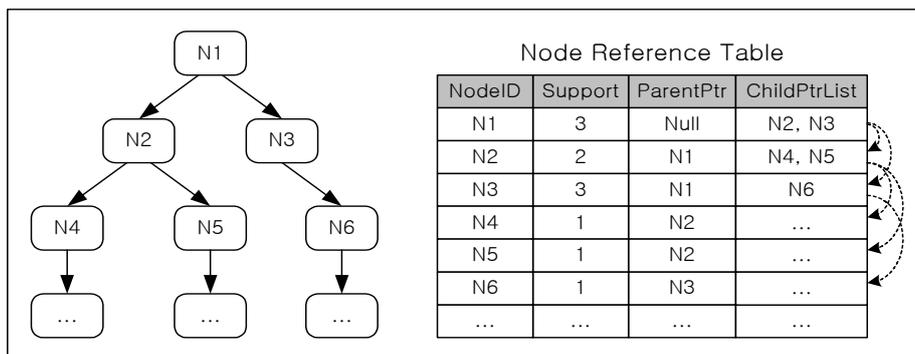
트랜잭션 데이터 변환 단계는 빈발 1-시퀀스 항목들이 각 객체별 이동 시퀀스 항목에 포함되었는지 검사하여 빈발 1-시퀀스 항목만을 남기고 나머지는 제거하여 트랜잭션 데이터 집합을 생성하는 단계이다. <표 2>의 이동 시퀀스 집합을 변환하면 <표 3>과 같다.

<표 2> 일반화된 이동 시퀀스 집합 예

OID	Sequence
MO1	<A5 A6 C8 C1>
MO2	<A5 C8 C9 C11>
MO3	<A5 A6 C1 C9>
MO4	<C8 C1 C9 C12> <C11>
MO5	<A6 C8 C1 C9 C11>
MO6	<A5 C8 C1 C9 C11>
MO7	<A4 C3 C8 C9>
MO8	<C8 C3 C12 C11>
MO9	<B1 A3 C10 C2 C11>

<표 3> 트랜잭션 데이터 집합 예

OID	Sequence
MO1	<A5 A6 C8 C1>
MO2	<A5 C8 C9 C11>
MO3	<A5 A6 C1 C9>
MO4	<C8 C1 C9 C12> <C11>
MO5	<A6 C8 C1 C9 C11>
MO6	<A5 C8 C1 C9 C11>
MO7	<C3 C8 C9>
MO8	<C8 C3 C12 C11>
MO9	<C11>



(그림 2) 이동 시퀀스 트리

**[단계 2] 이동 시퀀스 트리 구성 단계**

이동 시퀀스 트리 구성 단계는 앞선 단계에서 생성된 트

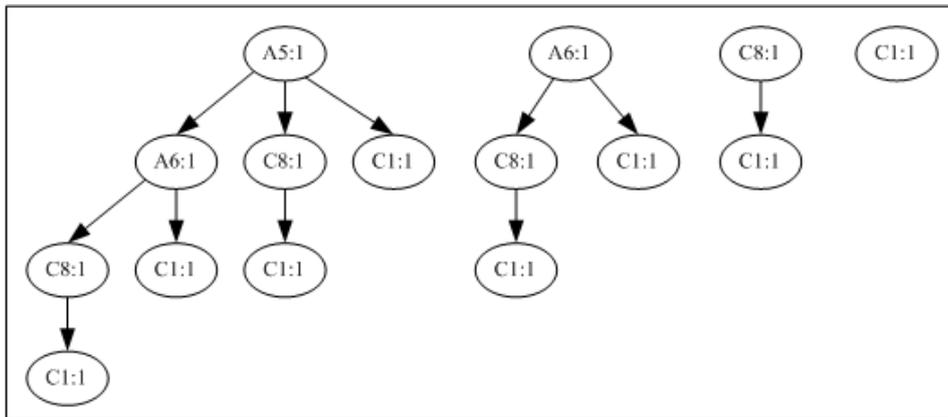
랜잭션 데이터 집합의 각 객체별 이동 시퀀스에 대한 모든 부분 시퀀스 집합을 생성하고, 각 부분 시퀀스들을 이용하여 이동 시퀀스 트리를 구성하는 단계로서, 각각의 시퀀스 트리는 개별적으로 하나의 노드 참조 테이블을 생성하여 각 노드들 간의 참조 정보와 지지도 정보를 유지한다. 하나의 이동 시퀀스에 대한 부분 시퀀스는 시퀀스 항목의 개수가 n 개 일 때, 공집합을 제외한  $2^n - 1$ 개의 부분 시퀀스를 생성한다. <표 3>에서 OID가 MO1인 객체의 이동 시퀀스에 대한 부분 시퀀스 집합을 추출하면 {<A5>, <A6>, <C8>, <C1>, <A5 A6>, <A5 A8>, <A5 C1>, <A6 C8>, <A6 C1>, <C8 C1>, <A5 A6 C8>, <A5 A6 C1>, <A5 C8 C1>, <A6 C8 C1>, <A5 A6 C8 C1>}, 총 15개의 부분 시퀀스들이 생성될 수 있다.

(그림 3)은 MO1 객체의 부분 이동 시퀀스 집합에 대해 시퀀스 트리를 구성한 예로, 트리의 구성은 먼저 각각의 1-부분 시퀀스 <A5>, <A6>, <C8>, <C1>을 루트노드로 생성하고, 각 노드의 지지도를 1씩 증가시킨다. 그 다음 2-시퀀스들은 각 루트노드의 하위에 노드를 생성하고, 새롭게 생성된 하위노드의 지지도만을 증가시킨다. 가령, 시퀀스 <A5 A6>의 경우 루트노드 A5의 하위노드로 A6을 생성하고 A6의 지지도를 1만큼 증가시킨다. 이는 A5와 A6의 참조 관계가 시퀀스 항목들 간의 순서를 의미하며, A6의 지지도

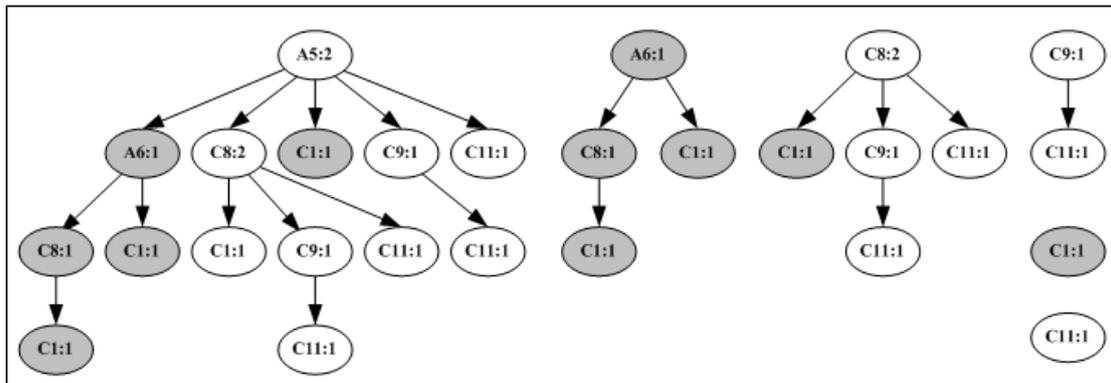
는 단순히 A6 항목에 대한 지지도가 아닌 <A5 A6> 시퀀스에 대한 지지도를 의미한다. 동일한 방식으로 나머지 2-시퀀스와 3-시퀀스, 4-시퀀스들에 대해 트리를 구성하고 지지도를 1만큼 증가시킨다. 모든 부분 시퀀스들에 대한 트리를 구성하면 (그림 3)과 같다.

다음 과정에서는 앞선 MO1에 대한 시퀀스 트리의 구성 방법과 동일한 방식으로 두 번째 객체인 MO2의 이동 시퀀스 <A5 C8 C9 C11>에 대한 부분 시퀀스 집합을 추출한 후 앞서 생성된 시퀀스 트리에 추가적으로 노드들을 생성하여 트리를 재구성한다. (그림 4)는 MO2의 이동 시퀀스에 대해 시퀀스 트리를 구성한 예로, 1-시퀀스들에 대한 트리 구성 시 <A5>와 <C8>은 앞서 이미 생성되었기 때문에 지지도만 1씩 증가시키고, <C9>와 <C11>은 새로운 루트노드로 생성한다. 또한 2-시퀀스들 중 <A5 C8>은 루트노드 A5의 하위노드로 C8이 이미 생성되어 있으므로 지지도를 1만큼 증가시키고, 나머지 시퀀스들은 시퀀스의 첫 번째 항목에 대응하는 루트노드들의 하위노드로 각각 두 번째 항목을 추가하여 트리를 구성한다. 마찬가지로 3-시퀀스들과 4-시퀀스들에 대해서도 트리를 구성한다.

앞선 MO1 객체에 대한 부분 시퀀스 집합의 추출과 달리 MO2 객체와 같이 하나 이상으로 구성된 시퀀스들에 대한 부분 시퀀스 집합은 각각의 시퀀스에 대해 먼저 부분 시퀀



(그림 3) MO1 객체의 이동 시퀀스에 대한 시퀀스 트리 구성 예



(그림 4) MO2 객체의 이동 시퀀스에 대한 시퀀스 트리 구성 예



〈표 5〉 최대 빈발 이동 패턴 집합

길이	부분 시퀀스
1	<A5> <A6> <C1> <C3> <C8> <C9> <C11> <C12>
2	<A5 A6> <A5 C1> <A5 C8> <A5 C9> <A5 C11> <A6 C1> <A6 C8> <A6 C9> <C1 C9> <C1 C11> <C8 C1> <C8 C9> <C8 C11> <C8 C12> <C9 C11> <C12 C11>
3	<A5 A6 C1> <A5 C1 C9> <A5 C8 C1> <A5 C8 C9> <A5 C8 C11> <A5 C9 C11> <A6 C1 C9> <A6 C8 C1> <C1 C9 C11> <C8 C1 C9> <C8 C1 C11> <C8 C9 C11> <C8 C12 C11>
4	<A5 C8 C9 C11> <C8 C1 C9 C11>

동 패턴 집합에는 약간의 중복된 패턴이나 다른 패턴의 부분집합에 해당하는 패턴이 존재한다. 따라서 빈발 패턴 집합으로부터 중복 및 부분집합 패턴들을 제거하여 최대 빈발 패턴의 집합을 추출하면 {<C3> <A5 A6 C1> <A5 C1 C9> <A5 C8 C1> <A6 C1 C9> <A6 C8 C1> <C8 C12 C11> <A5 C8 C9 C11> <C8 C1 C9 C11>}이다.

4.2 시공간 이동 패턴 탐사 알고리즘

STMPM\_MST 알고리즘은 이동 객체의 공간 속성과 시간 속성을 실제계의 공간지역과 시간영역으로 일반화하여 생성한 이동 시퀀스 집합으로부터 빈발 이동 패턴을 탐사하기 위한 알고리즘이다. (그림 6)의 STMPM\_MST 알고리즘

은 먼저, 이동 객체 데이터베이스로부터 *limitedDataSet* 함수를 통해 공간 및 시간 한정자  $C_s$ 와  $C_t$ 를 만족하는 이동 객체의 이력 데이터 집합을 추출하고, *SeqExtractor* 함수를 통해 연속되는 이동 객체의 이력 데이터들에 대해 시간 간격 조건 *max\_gap*을 적용하여 단위 이동 시퀀스를 추출한다. 추출된 이동 시퀀스 데이터 집합은 *Contains*와 *Durings* 함수를 통해 이동 시퀀스 데이터의 공간 속성과 시간 속성을 시공간 질의 분석을 통해 결정된  $L_{sk}$ 와  $L_{tk}$  수준으로 각각 일반화하고 중복된 시퀀스 항목을 요약하여 일반화-요약 이동 시퀀스 집합을 생성한다. 마지막으로 *FreqPatternExtractor* 함수는 일반화-요약 이동 시퀀스 집합과 최소지지도 *min\_sup*을 이용하여 트랜잭션 데이터 집합을 추출하고 이동 시퀀스

```

Input : D(Database), Cs(Spatial Constraint), Ct(Temporal Constraint),
        max_gap(Time Interval Constraint), Lsk(Spatial Level),
        Ltk(Temporal Level), min_sup(Threshold of Minimum Support)
Output : FrePatternSet

Procedure STMPM_MST(D, Cs, Ct, max_gap, Lsk, Ltk, min_sup)

Begin
    PrevA = null;

    // Set of History Data satisfied Constraint of Spatial and Temporal
    Drt = limitedDataSet (D, Cs, Ct);

    // Extracting Set of Moving Sequence from Drt
    Dseq = SeqExtractor (Drt, max_gap);

    // Generalizing Region to Spatial Property of Moving Sequence in Dseq
    DRgen = Contains (Dseq, PrevA, Lsk);

    // Generalizing Valid Time to Temporal Property of Moving Sequence in DRgen
    DTgen = Durings (DRgen, Ltk);

    // Frequent Moving Pattern Mining
    FreqPatternSet = FreqPatternExtractor (DTgen, min_sup);

    Return FreqPatternSet;
End
    
```

(그림 6) STMPM\_MST 알고리즘

트리의 구성을 통해 빈발 이동 패턴을 추출한다.

- 단위 이동 시퀀스 생성 알고리즘 *SeqExtractor* : 공간 및 시간 한정자  $C_s$ 와  $C_t$ 를 만족하는 이동 객체의 이력 데이터 집합  $D_n$ 을 객체 식별자와 유효시간을 기준으로 정렬하여 각 객체별 순차 리스트를 생성하고, 객체별 리스트의 두 연속되는 항목의 유효시간 값의 차가 시간 간격 제약 조건인  $max\_gap$ 을 초과하면 두 개의 시퀀스로 분리하여 단위 시퀀스를 생성하는 함수이다.

- 공간 속성 일반화 알고리즘 *Contains* : 이동 시퀀스 집합  $D_{seq}$ 에서 이동 객체의 위치 속성을 시공간 질의 분석을 통해 결정된 공간 수준  $L_{sk}$ 의 공간영역으로 일반화하기 위한 공간 연산 함수로, 객체의 위치값이 공간 개념 계층에서 공간 수준별 영역에 포함되었는지를 검사하여 이동 객체의 위치 속성을 공간영역으로 일반화한다.

- 시간 속성 일반화 알고리즘 *Durings* : 공간 일반화된 이동 시퀀스 집합  $D_{ngen}$ 에서 이동 객체의 시간 속성을 시공간 질의 분석을 통해 결정된 시간 개념 수준  $L_{tk}$ 의 시간 영역으로 일반화하기 위한 시간 간격 연산 함수로, 객체의 유효시간이 시간 개념 수준별 영역에 포함되는지 반복적으로 검사하여 상위 개념의 시간 영역으로 일반화한다.

- 빈발 이동 패턴 탐사 알고리즘 *FreqPatternExtractor* : 일반화-요약 이동 시퀀스 집합  $D_{Tgen}$ 과 사용자에 의해 지정된 최소지지도 임계치  $min\_sup$ 을 전달받아 빈발 이동 패턴을 추출하기 위한 함수로, 일반화-요약 이동 시퀀스 집합  $D_{Tgen}$ 으로부터 빈발 1-시퀀스 항목들로 구성된 트랜잭션 데이터 집합을 생성하고, 각 객체별로 부분 시퀀스 집합을 추출하여 이동 시퀀스 트리를 구성한다. 그 다음 트리를 구성하는 각 노드의 지지도와 최소지지도 임계치  $min\_sup$ 을 비교하여 작은 지지도를 갖는 노드들을 제거한 후 트리 탐색을 통해 빈발 이동 패턴을 추출한다.

### 5. 실험 및 성능 평가

제안된 빈발 이동 패턴 탐사 알고리즘 STMPM\_MST에 대한 성능 평가는 실험 데이터를 이용하여 기존의 시공간 이동 패턴 탐사 기법인 MPMine과 Apriori-MSP 알고리즘들을 대상으로 성능을 비교, 분석하였다. MPMine, Apriori-MSP 기법을 실험 대상으로 선택한 이유는 기존의 기법 순차 패턴 탐사에 가장 표준적인 방식인 Apriori 기법을 기반으로 하는 탐사 기법이고, 제안된 이동 시퀀스 트리 방식과 기존 후보 시퀀스 생성 방식과의 탐사 성능 비교를 중점적으로 수행하기 위해서 유사한 데이터 일반화 방법을 사용하는 기존의 탐사기법을 채택하여 실험하였습니다. 실험을 통해 각 알고리즘과 비교할 평가 기준은 최소지지도를 사용하여 패턴 탐사 수행 시간에 대한 효율성을 측정하였다.

실험을 위한 시공간 이동 패턴 탐사 시스템인 STMPMiner의 개발 환경은 Windows XP기반에서 개발도구로 Eclipse SDK 3.2.2와 JDK 1.6.0을 사용하였고, 이동 객체 데이터와 Geometry 데이터를 저장하기 위해서 Oracle 10g로 데이터베이스 시스템을 구축하였다. 최적 이동 패턴 추출 성능에 대한 실험 환경은 STMPMiner 시스템으로 Pentium IV 2.4GHz PC와 2GByte 메인 메모리를 사용하였고, 데이터베이스 시스템으로 Pentium IV 2.0GHz PC와 1GByte 메인 메모리를 사용하였다. 또한 실험을 위한 Geometry 데이터로 서울시 행정 구획 데이터와 도로 네트워크 데이터를 사용하였고, 이동 객체 데이터로 서울시의 도로 네트워크 상에서 택시들의 운행 기록을 추위하여 이력 데이터를 생성하였다. 추위된 이력 데이터는 도로 네트워크 상에서의 교차점이나 분기점, 종료점 등을 노드로 설정하여 각 노드에서 이동 객체의 위치 정보를 샘플링하였다. 다음 <표 6>은 성능 평가 실험에 이용되는 이력 데이터의 유형이고, <표 7>은 실험 데이터에 대한 특성을 나타낸다. 데이터 집합의 이름에서 N은 이동 객체 수, D는 이동기간, G는 샘플링 위치를 의미한다.

이동 객체 수에 따른 각 최소지지도별 빈발 패턴 탐사 시간은 시간 한정 크기를 10시간으로 고정하고 최소지지도가

<표 6> 성능 평가를 위한 실험 데이터 유형

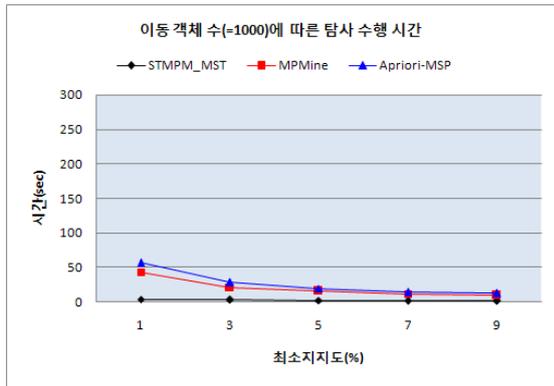
MOID	NODE		Valid Time
	x	y	
15440001	126.962847	37.533317	2006/10/14/11/12/02
15440001	126.970583	37.536738	2006/10/14/11/12/53
15440001	126.970708	37.541675	2006/10/14/11/13/50
15440001	127.003377	37.502183	2006/10/14/11/14/01
...	...	...	...

<표 7> 실험 데이터 특성

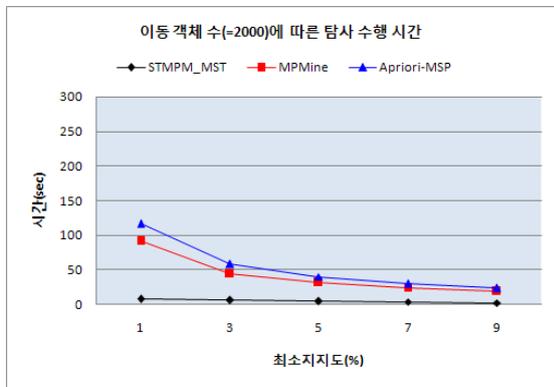
데이터 집합 이름	이동 객체 수(N)	이동기간(D)	실험데이터 크기
N4000-D1-Gnode	4000	1일	58.9 MB

각각 1%, 3%, 5%, 7%, 9% 일 때, 이동 객체의 수를 1000 개, 2000개, 3000개로 변화시키면서 패턴 탐사를 위한 연산 시간을 측정하였다. 다음 (그림 7), (그림 8), (그림 9)는 이동 객체 수를 각각 1000개, 2000개, 3000개로 고정했을 때 최소 지지도 변화율에 따른 탐사 수행 시간을 측정된 결과이다.

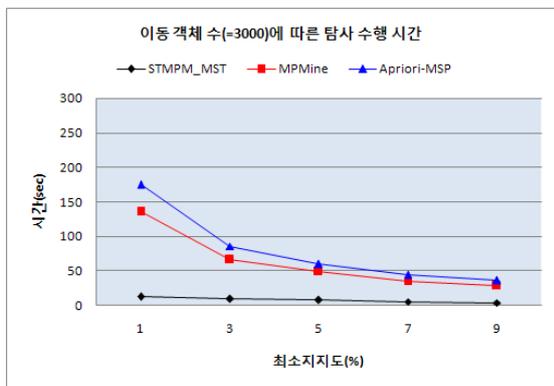
빈발 패턴 탐사 수행 시간은 데이터 전처리 시간과 빈발 패턴 추출 시간의 합으로 구성된다. STMPM\_MST의 전처리 시간은 MD 데이터 정렬, 단위 시퀀스 추출, 공간/시간 속성 일반화, 데이터 요약 등에 걸리는 시간의 합이다. MPMine은 STMPM\_MST에 비해 시간속성의 일반화 시간



(그림 7) 탐사 시간 : 이동 객체 수=1000



(그림 8) 탐사 시간 : 이동 객체 수=2000



(그림 9) 탐사 시간 : 이동 객체 수=3000

이 없기 때문에 전처리 시간이 감소할 수 있지만, Apriori-MST는 패턴 탐사 문제의 특성 상 단위 시퀀스를 추출하는데 걸리는 시간은 없지만 유효시간 필터링 및 유효소요시간 산정시간이 단위 시퀀스 추출 시간보다 많은 시간이 소요되기 때문에 전처리 시간이 가장 많이 소요되었다. 하지만 STMPM\_MST, MPMine, Apriori-MSP 알고리즘은 데이터 전처리 시간에서 빈발 패턴 추출 시간에 비해 큰 차이를 보이지 않았다.

실험 결과를 보면 STMPM\_MST의 경우 최소지지도 변화율에 따라 큰 폭의 차이가 없이 거의 일정하게 탐사 수행 시간이 감소하지만, MPMine과 Apriori-MSP는 최소지지도가 낮은 경우 수행시간이 많이 걸리며 지지도가 높아질수록 큰 폭으로 수행 시간이 감소하는 결과를 보인다. 이는 최소 지지도가 낮을수록 MPMine과 Apriori-MSP에서 패턴 탐사를 위해 생성하는 후보 시퀀스의 수가 증가하면서 찾은 데이터베이스 스캔을 필요로 하고, 각 후보 시퀀스와의 비교 연산 시간이 증가하여 큰 폭으로 수행 시간이 증가하기 때문이다.

빈발 패턴 추출 시간에서 MPMine과 Apriori-MST 알고리즘은 패턴 탐사에 Apriori 방식을 적용하고 있어 후보 시퀀스 집합을 생성하고 최소지지도를 계산하는 때 패스마다 트랜잭션 데이터베이스를 스캔하기 때문에 많은 탐사 시간이 소요되었다. 하지만 MPMine의 경우 트랜잭션 데이터베이스 스캔 시간을 감소시키기 위한 해시 테이블을 사용함으로써 Apriori-MST 알고리즘에 비해 후보 시퀀스의 수를 감소시켜 최소지지도 계산 시간을 줄여 전체 빈발 패턴 추출 시간을 감소시킬 수 있었다.

하지만 STMPM\_MST의 경우 최소지지도의 변화에 있어 일정한 탐사 시간을 유지하는 이유는 빈발한 후보 1-시퀀스 집합으로부터 후보 k-시퀀스 집합을 생성하지 않고 빈발한 1-시퀀스 항목으로 구성된 변형된 트랜잭션 시퀀스 데이터로부터 이동 시퀀스 트리를 생성하는 방식을 통해 지지도를 계산하여 빈발 패턴을 추출하기 때문이다. 이러한 패턴 추출 시간은 후보 1-시퀀스 집합으로부터 더 이상 후보 k-시퀀스 집합을 생성할 수 없을 때까지 매 패스마다 이루어지는 트랜잭션 데이터베이스 스캔 시간과 최소지지도 비교 시간의 합보다 상대적으로 훨씬 적게 소요되어 탐사 수행 시간을 감소시킬 수 있었다.

## 6. 결론 및 향후 연구과제

최근 이동 객체의 동적인 위치나 이동성에 기반하여 다양한 응용 분야에서 활용할 수 있는 새로운 위치 기반 서비스를 개발하고자 하는 노력이 증가하고 있으며, 사용자의 동적인 위치에 기반한 특화되고 개인화된 서비스를 제공하고 자 시공간 이동 패턴 마이닝 기술의 사용이 불가피한 상황이다. 하지만 현재까지 많은 패턴 탐사 기법들이 연구되고 개발되었지만 아직까지 패턴 탐사 시 발생하는 많은 시스템 부하로 인해 대용량급의 시스템을 구축해야만 활용될 수

있는 단점이 있다.

이에 본 논문에서는 방대한 양의 이동 객체 데이터에 대한 패턴 탐사를 수행함에 있어 저사양의 시스템에서도 원활하게 패턴 탐사를 수행할 수 있도록 탐사 수행 시간 및 패턴 탐사 시 소요되는 메모리량을 최소화하기 위한 새로운 빈발 패턴 탐사 기법인 STMPM\_MST 기법을 제시하였다. 제안된 기법은 시공간 이동 데이터 집합으로부터 순차적이고 주기적인 빈발한 이동 패턴을 탐사하는데 있어 이동 시퀀스 트리를 구성하여 패턴 탐사를 수행함으로써 데이터베이스에 대한 접근을 최소화하여 탐사 수행 시간을 감소시키고, 상세 수준의 이력 데이터를 실세계의 공간 및 시간 개념으로 일반화하여 탐사 시 소요되는 메모리량을 최소화하도록 하였다. 시공간 빈발 패턴 탐사를 위한 STMPM\_MST 알고리즘은 데이터 전처리 과정과 빈발 이동 패턴 탐사 과정을 거쳐 패턴 탐사를 수행하도록 구성하였다. 먼저 전처리 과정에서는 효율적인 패턴 탐사를 위해 이동 객체의 이력 데이터 집합을 시간 간격 제약 조건을 적용하여 단위 이동 시퀀스로 생성하고, 생성된 시퀀스들에 대해 공간 및 시간 개념 계층에 따라 데이터를 일반화 및 요약하였다. 또한, 빈발 이동 패턴 탐사 과정에서는 일반화-요약 이동 시퀀스 집합을 빈발 1-시퀀스 항목들만으로 구성된 트랜잭션 데이터 집합으로 변환하고, 각 객체별 트랜잭션 데이터에 대해 부분 시퀀스 집합을 추출한 후 이동 시퀀스 트리를 구성하여 빈발 이동 패턴을 탐사하였다.

실험 및 성능 평가에서는 제안된 STMPM\_MST 알고리즘과 기존의 탐사 기법인 MPMine, Apriori-MSP 알고리즘을 대상으로 이동 객체 수의 변화에 따른 각 최소지지도별 탐사 수행 시간을 비교하였다. 실험 결과 STMPM\_MST 알고리즘은 MPMine과 Apriori-MSP 알고리즘의 패턴 탐사 수행 시간을 각각 82.6%~91.4%, 86.3%~93.4% 감소시켜 두 알고리즘에 비해 훨씬 좋은 성능을 보였다. 이는 이동 시퀀스 트리의 구성을 통해 빈발 패턴을 탐사하는 STMPM\_MST 알고리즘이 데이터베이스에 대한 접근을 최소화하여 전반적인 탐사 수행 시간을 감소시킴으로써 효과적인 패턴 탐사를 수행할 수 있었다.

본 논문의 연구 결과들은 이동 객체의 빈발한 이동 패턴을 추출하여 특정 분야에 적합한 위치 기반 서비스로 활용 가능하며, 임의의 네트워크 안에서 사용자의 이동성 예측을 위한 이동성 패턴을 추출하는 UMP(User Mobility Pattern) 마이닝 기법에 적용가능하다. 또한 다양한 이동 객체들의 공통적인 이동 패턴을 분석하여 그 패턴에 따라 수집자/분배자 배치, 특정 지점들 간의 최적 이동 경로 제공, 정해진 시간 내에서의 효율적인 스케줄링 경로 탐색과 같은 시간 및 공간 제약 조건을 갖는 빈발 패턴을 탐사하는 문제에 적용가능하다.

향후 연구과제는 이동 객체의 위치 이력에 대한 정보뿐만 아니라 이동 객체의 행위 정보(속도, 방향, etc)와 이동 객체가 특정 위치에 머무른 기간을 고려한 패턴 탐사 방법에 대한 연구가 수행되어야 한다. 또한, 일반적으로 빈발하게 발생하는 이동 패턴뿐만 아니라 외부적인 요인이나 특정

한 상황 발생에 따른 규칙적인 이동 객체의 움직임 탐사하기 위한 탐사 방법의 개발이 필요하다.

## 참 고 문 헌

- [1] D. O. Kim, H. K. Kang, D. S. Hong, J. K. Yun and K. J. Han, "STMPE : An Efficient Movement Pattern Extraction Algorithm for Spatio-temporal Data Mining", in proc. on International Conference on Computational Science and Its Applications(ICCSA), pp.259-269, 2006.
- [2] N. Mamoulis, H. Cao, G. Kollios, M. Hadjieleftheriou, Y. Tao and D. W. Cheung, "Mining, Indexing and Querying Historical Spatio-Temporal Data", in proc. on the International Conference on Knowledge Discovery and Data Mining, 2004.
- [3] H. Cao, N. Mamoulis and D. W. Cheung, "Mining Frequent Spatio-Temporal Sequential Patterns", in proc. on the 5th IEEE International Conference on Data Mining(ICDM), pp.82-89, 2005.
- [4] Y. Huang, L. Zhang and P. Zhang, "Finding Sequential Patterns from a Massive Number of Spatio-Temporal Events", SDM, SIAM, 2006.
- [5] J. W. Lee, O. H. Paek and K. H. Ryu, "Temporal Moving Pattern Mining for Location-Based Service", The Journal of Systems and Software, Vol.73, 2004.
- [6] G. Yavas, D. Katsaros, O. Ulusoy and Y. Manolopoulos, "A Data Mining Approach for Location Prediction in Mobile Environmensts", Data & Knowledge Engineering, Vol.54, pp.121-146, 2005.
- [7] J. Allen, "Maintaining Knowledge about Temporal Intervals", Comm. of the ACM, Vol.26, No.11, 1983.
- [8] 백옥현, "위치 기반 서비스를 위한 이동 객체의 시간 패턴 탐사 기법", 충북대학교 대학원, 석사학위논문, 2002.
- [9] 이준욱, "위치 기반 서비스를 위한 이동 객체의 시간 패턴 탐사", 한국정보과학회 논문지, 제29권, 제5호, 2002.
- [10] 이준욱, "지식 탐사 프레임워크 기반의 시공간 이동 패턴 탐사 기법", 충북대학교 대학원, 박사학위논문, 2003.
- [11] 한선영, "시공간 이동 시퀀스 패턴 마이닝 기법", 이화여자대학교 대학원, 석사학위논문, 2006.
- [12] 박지웅, "시공간 이동 패턴 추출을 위한 효율적인 알고리즘", 건국대학교 대학원, 박사학위논문, 2006.
- [13] 고 현, 김광중, 이연식, "최적 경로 서비스 지원을 위한 이동 객체의 이동 패턴 탐사 알고리즘", 한국정보처리학회 추계학술발표대회논문집, 제7권, 제2호, 2006. 11.
- [14] 고 현, 김광중, 이연식, "이동 객체의 패턴 마이닝을 위한 위치 일반화 방법", 한국정보처리학회 추계학술발표대회논문집, 제7권 제2호, 2006. 11.
- [15] 고현, 김광중, 이연식, "이동 객체의 패턴 탐사를 위한 시공간 데이터 일반화 기법", 한국정보과학회 한국컴퓨터종합학술대회논문집(C), 제34권, 제1호, 2007. 06.



**이 연 식**

e-mail : yslee@kunsan.ac.kr  
1982년 전남대학교 전자계산학과(학사)  
1984년 전남대학교 전자계산학과(이학석사)  
1994년 전북대학교 전산응용공학과(공학박사)  
1995년~1997년 군산대학교 교무부처장

1997년~1998년 University of Missouri 교환교수  
1999년~2001년 군산대학교 전자계산소 소장  
2004년~2005년 Ohio State University 교환교수  
1986년~현 재 군산대학교 컴퓨터정보공학과 교수  
관심분야 : 언어 번역기 이론, 능동 객체지향시스템, 지능형 에이전트, 마이닝 에이전트



**고 현**

e-mail : khyun@kari.re.kr  
2001년 군산대학교 컴퓨터정보학과(학사)  
2003년 군산대학교 컴퓨터정보학과(이학석사)  
2007년 군산대학교 컴퓨터정보학과(이학박사)

2008년~현 재 한국항공우주연구원 프로젝트연구원  
관심분야 : 에이전트, 이동객체시스템, 시공간데이터베이스, 데이터마이닝