

STMP/MST와 기존의 시공간 이동 패턴 탐사 기법들과의 성능 비교[☆]

A Comparison of Performance between STMP/MST and Existing Spatio-Temporal Moving Pattern Mining Methods

이 연 식*
Yonsik Lee

김 은 아**
Euna Kim

요 약

시공간 이동 패턴 탐사는 특성상 방대한 시공간 데이터의 분석 및 처리 방법에 따라 패턴 탐사의 성능이 좌우된다. 기존의 시공간 패턴 탐사 기법들(1-10)이 가진 패턴 탐사 수행 시간이나 패턴 탐사 시 사용되는 메모리량이 증가하는 문제를 해결하기 위해 일부 기법에서 몇 가지 방법을 제시하였으나 아직 미비한 실정하다.

이에 선행 연구로 방대한 시공간 이동 데이터 집합으로부터 순차적이고 주기적인 빈발 이동 패턴을 효과적으로 추출하기 위한 STMP/MST 탐사 기법(11)을 제안하였다. 제안된 기법은 해시 트리 기반의 이동 시퀀스 트리를 생성하여 빈발 이동 패턴을 탐사함으로써 탐사 수행 시간을 최소화하고, 상세 수준의 이력 데이터들을 실세계의 의미있는 시간 및 공간영역으로 일반화하여 탐사 시 소요되는 메모리량을 감소시킬 수 있다.

본 논문에서는 이러한 STMP/MST 탐사 기법의 효율성을 검증하기 위해서 탐사 대상 데이터양과 최소지지도를 기준으로 기존의 시공간 패턴 탐사 기법들과 탐사 수행 성능을 비교하고 분석한다.

Abstract

The performance of spatio-temporal moving pattern mining depends on how to analyze and process the huge set of spatio-temporal data due to the nature of it. The several method was presented in order to solve the problems in which existing spatio-temporal moving pattern mining methods(1-10) have, such as increasing execution time and required memory size during the pattern mining, but they did not solve properly yet. Thus, we proposed the STMP/MST method(11) as a preceding research in order to extract effectively sequential and/or periodical frequent occurrence moving patterns from the huge set of spatio-temporal moving data.

The proposed method reduces patterns mining execution time, using the moving sequence tree based on hash tree. And also, to minimize the required memory space, it generalizes detailed historical data including spatio-temporal attributes into the real world scopes of space and time by using spatio-temporal concept hierarchy. In this paper, in order to verify the effectiveness of the STMP/MST method, we compared and analyzed performance with existing spatio-temporal moving pattern mining methods based on the quantity of mining data and minimum support factor.

☞ Keywords: 시공간 패턴 탐사(Spatio-temporal Pattern Mining), 빈발 이동 패턴(Frequent Moving Pattern), 이동 시퀀스 트리(Moving Sequence Tree), 성능 평가(Performance Evaluation)

1. 서 론

이동 객체는 시간의 흐름에 따라 공간 및 비공간 속성이 끊임없이 변화하는 시공간 객체(spatio-temporal object)이다. 현재까지 시간 및 공간 속성을 갖는 이동 객체에 관한 많은 연구가 수행되고 있으며, 최근 이동 객체의 동적인 위치 변

* 정 회 원 : 군산대학교 컴퓨터정보공학과 교수
yslee@kunsan.ac.kr(교신저자)

** 정 회 원 : 한국폴리텍 II인천캠퍼스 컴퓨터정보과 교수
ekim@kopo.ac.kr

[2008/12/08 투고 - 2008/12/13 심사 - 2009/04/07 심사완료]

☆ 본 연구는 한국과학재단 특정기초연구(R01-2007-20989-0) 및 중소기업청의 산학 공동기술개발사업의 지원으로 수행되었음

화에 따른 특성 분석을 통해 유용한 정보를 추출하여 보다 다양한 응용 분야에서 활용할 수 있는 새로운 위치 기반 서비스(LBS : Location Based Service)[12-15]를 개발하고자 시공간 패턴 탐사에 대한 관심과 연구가 날로 증가하고 있다. 시공간 이동 패턴 탐사는 연속되는 시간영역에서 이동하는 객체의 패턴들 중 객체의 위치 변화에 따른 시간 규칙이나 공간 규칙, 시공간 규칙을 탐사하기 위한 기술이다. 이동 패턴이란 연속되는 시간 영역에서 빈번하게 발생하는 이동 객체의 공간 이동 패턴을 의미한다. 현재까지 이러한 이동 패턴을 탐사하기 위한 많은 기법들이 제안되었으며, 대부분 연속적인 시간의 흐름에 따라 발생하는 객체의 이동 패턴들 중 주기적인 패턴[1,2]이나 순차적인 패턴[3-10]의 탐사를 목적으로 하고 있다. 하지만 기존의 시공간 패턴 탐사 기법은 패턴 탐사 시 후보 이동 시퀀스를 검증하거나 지지도 계산을 위해 빈번하게 데이터베이스를 스캔하는 경우 탐사 수행 시간이 증가하는 문제가 있다. 또한, 최소지지도가 낮아 후보 시퀀스 집합이 커지거나 탐사 대상 데이터양이 많은 경우에도 탐사 수행 시간뿐만 아니라 탐사 시 소요되는 메모리양이 급격히 증가하는 문제도 있다. 이에 따라 일부 기법들은 탐사 대상 데이터양에 따른 탐사 수행 시간 및 메모리 소요량을 감소시키기 위해서 데이터 일반화 방법을 적용하였고[1,5,7-10], 일부는 잦은 데이터베이스의 스캔에 따른 탐사 수행 시간을 감소시키기 위해서 빈발 1-패턴의 조합 또는 일반화된 이동 시퀀스로부터 패턴 트리를 생성하여 지지도 계산 및 후보 이동 시퀀스에 대한 검증을 수행하였다[1,2]. 하지만 이러한 기법들은 일반화 방법에 있어 실세계의 시간 및 공간에 대한 고려없이 사용자 정의에 따라 물리적인 형태로 일반화를 수행하기 때문에 일반화된 데이터로부터 실세계의 공간 및 시간에 대한 의미를 얻는 것이 어렵고, 이동 데이터의 양에 따라 패턴 트리를 생성하고 유지하는 비용이 커지는 문제가 있다.

이에 선행 연구로서 방대한 시공간 이동 데이터 집합으로부터 순차적이고 주기적인 빈발 이동

패턴을 효율적으로 추출하기 위한 새로운 시공간 이동 패턴 탐사 기법인 STMP/MST (Spatial-Temporal Moving Pattern Using Moving Sequence Tree) 기법을 제안하였다. 제안된 기법은 이동 객체의 이력 데이터들로부터 빈발 이동 패턴을 탐사하는데 있어 시간 및 공간 속성을 가진 상세 수준의 이력 데이터들을 공간 및 시간 개념 계층을 이용하여 의미있는 시간 및 공간영역으로 일반화하고, 해시 트리 기반의 이동 시퀀스 트리(Moving Sequence Tree)를 생성하여 빈발 패턴을 추출함으로써 탐사 수행 시간 및 소요 메모리양을 최소화할 수 있는 기법이다[11].

본 논문에서는 데이터 일반화와 이동 시퀀스 트리를 통해 패턴 탐사를 수행하는 STMP/MST 탐사 기법의 효율성을 검증하기 위해서 동일한 실험 조건에서 기존의 시공간 패턴 탐사 기법들과 탐사 수행 성능을 분석한다. 실험 조건은 탐사 대상 데이터양과 최소지지도를 기준으로 각각의 기준 요소를 일정하게 변화시키며 탐사 수행 시간을 측정하여 비교한다.

본 논문의 구성은 2장에서 관련연구로서 기존의 이동 패턴 탐사 기법에 대해 고찰한다. 3장에서는 데이터 일반화와 이동 시퀀스 트리의 생성을 통해 시공간 빈발 이동 패턴을 탐사하는 과정을 보이고, 4장에서는 이러한 탐사 방법에 대한 알고리즘을 설계한다. 5장에서는 실험을 통해 제안된 탐사 기법의 성능을 평가하고, 마지막 6장에서는 결론 및 향후 연구 과제를 제시한다.

2. 관련연구

이동 객체의 시공간 이동 패턴 탐사는 연속되는 시간영역에서 이동하는 객체의 패턴들 중위치 변화에 따른 독특한 경향이나 반복적이고 공통적인 의미있는 패턴을 탐사하기 위한 기술이다. 이동 패턴이란 연속되는 시간 영역에서 빈번하게 발생하는 이동 객체의 공간 이동 패턴을 의미한다. 현재까지 이러한 객체의 이동 패턴을 탐사하기 위한 많은 기법들이 제안되었다. 기존의 시공

간 이동 패턴 탐사 기법들은 대부분 주기적인 빈발 이동 패턴[1,2]과 순차적인 빈발 이동 패턴 [3-10]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 메모리량이 증가하여 탐사 성능이 저하되는 문제가 있다. 다음은 기존의 시공간 이동 패턴 탐사 기법들이 빈발한 이동을 패턴을 탐사하는데 있어 효율적인 탐사를 수행하는지 판별하기 위해서 패턴 탐사 방법 및 후보 시퀀스 검증 방법, 데이터베이스 접근 횟수, 데이터 일반화 지원여부 등의 4가지 측면에서 각 기법들의 성능을 비교하였다.

(표 1) 시공간 이동 패턴 탐사 기법들의 탐사 방법

기법	탐사방법	후보 시퀀스 검증 방법	데이터 베이스 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)

(표 1)에서 STPMine1[2]와 UMP[6], Apriori-MSP[10], MP&GMP[5,7,8], MPMine[9]는 Apriori 계열의 알고리즘을 변형하여 사용하기 때문에 빈발 항목 탐사 시 후보 항목 집합의 크기가 클수록 지지도 계산 및 새롭게 생성된 후보 항목들의 검증을 위해서 빈번하게 데이터베이스를 반복 스캔함으로써 탐사 수행 시간이 증가하는 문제를 가지고 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[1,16]는 각각 개별적인 패턴 트리를 생성하여 패턴 탐사를 수행함으로써 데이터베이스에 대한 빈번한 접근을 감소시켜 탐사 수행 시간을 최소화하도록 하였으나 유효시간 기준의 샘플링 방식을 통해 추출한 이력 데이터를 대상으로 탐사를 수행하기 때문에 일반적으로 위치 기준의 샘플링 방식을 통해 추출된 이력 데이터에 대해서는 변환 과정이 필요하며, 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 시간 속성 및 공간 속성 모두에 대해 각각 일반화를 수행하였다. 하지만 이들 기법에서 적용한 일반화 방법은 실세계의 시간 및 공간 개념에 대한 고려없이 단순히 사용자 정의에 따라 물리적인 형태로 시간영역과 공간영역을 분할하여 각 영역 대표값으로 일반화를 수행하였기 때문에 패턴 탐사를 통해 추출되는 지식을 활용하기 위해서는 지식 응용 분야별로 사용하는 개념적인 시간영역과 공간영역으로의 매핑이 필요하다. 또한, MP와 MPMine은 실세계의 공간지역에 대한 공간 개념 계층을 형성하여 이동 객체의 공간 속성에 대한 일반화만 수행하고 시간 속성에 대한 일반화는 수행하지 않았기 때문에 지식 응용 분야에서 적용되고 있는 시간 개념에 대한 시간영역으로의 일반화가 요구된다.

지금까지 고찰한 기존의 연구들에 대한 문제점을 해결하기 위하여 [11]에서는 최소지지도와 이동 객체 수 등에 영향을 받지 않고 일정한 탐사 수행 시간을 유지하며 기존의 탐사 기법보다 더 적은 탐사 시간을 소요하는 STMP/MST 탐사 기

법을 제안하였다. 본 논문에서는 제안된 STMP/MST 기법의 탐사 수행 시간에 대한 효율성을 검증하기 위해서 기존의 패턴 탐사 기법인 MPMine[9]과 Apriori-MSP[10] 기법을 대상으로 최소지지도, 데이터의 시간 한정 크기, 이동 객체 수를 변화시키며 탐사 시간에 대한 성능을 비교하고 분석한다.

3. 시공간 빈발 이동 패턴 탐사 과정

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. 기존의 시공간 패턴 탐사 기법은 이동 객체의 공간 정보가 연속적으로 변화하는 순차패턴 뿐만 아니라 주기적 또는 반복적으로 발생하는 주기패턴도 탐사할 수 있으나 관련연구에서 고찰한 바와 같이 패턴 탐사 수행 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 문제 정의(11)

이동 객체 데이터베이스 MD, 사용자가 지정한 최소지지도 min_sup , 공간 영역 간의 시간 간격 제약 조건 max_gap 이 주어졌을 때, 최소지지도를 만족하는 모든 빈발 이동 시퀀스를 탐색하는 것이다.

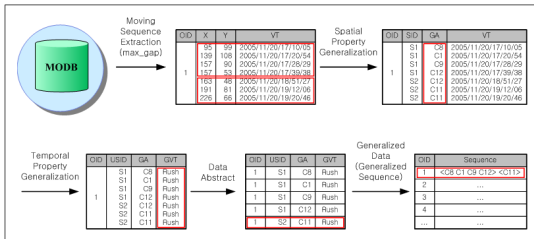
[정의 1]에서 최소지지도 min_sup 는 시퀀스 s 가 빈발한 것으로 평가되기 위해 만족해야 하는 지지도의 하한값이다. 시간 간격 제약 조건은 이동 객체의 이력 데이터 집합으로부터 패턴 탐사를 위한 트랜잭션 데이터 집합을 추출하기 위해서 연속되는 객체의 이동 이력을 한번의 이동에 대한 단위 이력, 즉 단위 이동 시퀀스로 분리하기 위한 제약 조건이다.

3.1 데이터 전처리 단계

시공간 빈발 이동 패턴 탐사 과정에서는 [정의 1]을 기반으로 이동 객체의 이력 데이터를 변환하는 전처리 과정과 빈발한 이동 패턴을 탐사하는 패턴 탐사 과정을 거쳐 효율적으로 빈발 이동 패턴을 추출한다. 전처리 과정에서는 단위 이동 시퀀스 생성, 단위 이동 시퀀스의 공간 및 시간 속성 일반화, 일반화된 데이터의 요약 과정을 거쳐 패턴 탐사 수행에 적합한 형태로 데이터를 변환한다[11]. 단위 이동 시퀀스의 생성은 하나의 이동 시퀀스를 구성하는 위치들 간에 최대 시간 간격에 대한 제약 조건을 두어 이를 만족할 경우에만 단위 이동 시퀀스로 생성한다. 시간 간격 제약 조건 max_gap 은 이동 객체의 연속적인 이동 이력 중 특정 영역에서 인접한 다른 영역으로의 이동 시간 $t_j - t_{j-1}$ 이 가질 수 있는 최대 시간 간격으로, 만약 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[17].

이동 객체의 공간 및 시간 속성 일반화는 *Contains*과 *During* 연산을 이용하여 일반화를 수행한다. *Contains* 연산은 공간 객체들 간의 포함 관계를 정의한 연산으로, 이동 객체의 위치값과 공간영역 간의 포함 여부를 고려하여 객체의 위치 속성을 실제계의 공간 개념 계층에 따라 공간영역으로 일반화한다[18,19]. 또한, *During* 연산은 시간 영역들 간의 포함 관계를 정의한 연산으로, 객체가 공간상에서 이동할 때 이산적으로 샘플링된 유효시간과 시간영역 간의 관계를 분석하여 객체의 시간 속성을 지식 응용 분야의 시간 개념 계층에 따라 시간영역으로 일반화한다[20]. 일반화된 데이터의 요약은 연속된 시퀀스 항목들 중 공간 속성과 시간 속성이 동일한 경우 하나의 시퀀스 항목으로 요약한다. 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[11]. (그림 1)은 이동 객체의 이력 데이터를 전처리 과정을 통해 일반화-요약된 시퀀스 집합으로 추출하는 과정이다.



(그림 1) 이동 객체의 이력 데이터 전처리 과정

3.2 빈발 이동 패턴 탐사 단계

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[11]. 패턴 탐사 과정은 Apriori 계열의 알고리즘과 같이 트랜잭션 데이터베이스를 여러 번 스캔하여 후보 시퀀스 집합에 대한 지지도 계산을 통해 빈발 패턴을 탐사하는 대신 이동 시퀀스 트리를 구성하는 과정을 통해 빈발 패턴을 탐사한다. 기존의 Apriori 계열 알고리즘에 의한 패턴 탐사는 빈번한 데이터베이스 스캔으로 탐사 수행 시간이 많이 걸리는 문제가 있으므로 이동 시퀀스 트리의 구성을 통해 총 2회의 데이터베이스에 대한 접근만을 허용함으로써 보다 효율적으로 빈발 이동 패턴을 추출할 수 있다. 데이터베이스 스캔은 빈발 1-시퀀스 집합을 추출하는 경우와 각 시퀀스 데이터를 읽어들이며 이동 시퀀스 트리를 구성하는 경우에만 이루어진다. 최대 빈발 패턴을 탐사하기

위한 각 과정은 다음과 같다[11].

[단계 1] 빈발 1-시퀀스 집합 발견 단계
트랜잭션 데이터베이스로부터 패턴 탐사를 수행할 일반화-요약된 이동 시퀀스 집합을 읽어들이며 빈발 1-시퀀스 집합을 생성한다.

[단계 2] 트랜잭션 데이터 변환 단계
빈발 1-시퀀스 집합이 각 이동 객체별 이동 시퀀스에 포함되었는지 검사하여 빈발 1-시퀀스 항목들을 포함하고 있는 이동 시퀀스의 항목만을 남기고 나머지는 제거하여 새로운 이동 시퀀스 집합을 생성한다. 이 과정을 통해 각 이동 시퀀스의 항목들이 자주 발생하는 항목들의 리스트만으로 표현된다.

[단계 3] 이동 시퀀스 트리 구성 단계
하나씩 이동 시퀀스를 입력으로 받아 이동 시퀀스의 모든 부분 시퀀스 집합을 생성하고, 각 부분 시퀀스들을 이용하여 이동 시퀀스 트리의 구성한 후 노드 참조 테이블에 각 노드들 간의 참조 정보와 지지도를 증가시킨다. 이 때, 이미 부분 시퀀스에 대한 이동 시퀀스 트리의 구성이 이루어진 경우 노드 참조 테이블에서 해당 시퀀스에 대한 지지도만을 증가시킨다.

[단계 4] 빈발 이동 패턴 탐사 단계
모든 이동 시퀀스 집합에 대한 이동 시퀀스 트리 생성이 완료된 후 시퀀스 트리를 탐색하여 노드 참조 테이블의 지지도 필드값을 사용자가 지정한 최소지지도 임계값과 비교하여 빈발 패턴을 추출한다.

4. 시공간 빈발 이동 패턴 탐사 알고리즘

시공간 빈발 이동 패턴 탐사 알고리즘인 STMP/MST는 이동 객체의 공간 속성과 시간 속성을 실세계의 공간지역과 시간영역으로 일반화하여 생성된 이동 시퀀스 집합으로부터 빈발 이동 패턴을 탐사하기 위한 알고리즘이다. (그림 2)의 STMP/MST 알고리즘은 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*를 적용하여 이동 시퀀스 데이터를 생성한다. 생성된 이동 시퀀스 데이터 집합은 *Contains*[18,19]와 *Durings*[20] 함수

Input : D (*Database*), C_s (*Spatial Constraint*), C_t (*Temporal Constraint*)
 max_gap (*Time Interval Constraint*), Ls_k (*Spatial Level*),
 Lt_k (*Temporal Level*), min_sup (*Thershold of Minimum Support*)
Output : *FreqPatternSet*

Procedure $STMP(D, C_s, C_t, max_gap, Ls_k, Lt_k, min_sup)$

Begin

$PrevA = null;$

//Set of History Data satisfied Constraint of Spatial and Temporal

$D_{rt} = limitedDataSet(D, C_s, C_t);$

// Extracting Set of Moving Sequence from D_{rt}

$D_{seq} = SeqExtractor(D_{rt}, max_gap);$

// Generalizing Region to Spatial Property of Moving Sequence in D_{seq}

$D_{Rgen} = Contains(D_{seq}, PrevA, Ls_k);$

// Generalizing Valid Time to Temporal Property of Moving Sequence in D_{Rgen}

$D_{Tgen} = Durings(D_{Rgen}, Lt_k);$

//Frequent Moving Pattern Minging

$FreqPatternSet = FreqPatternExtractor(D_{Tgen}, min_sup);$

Return FreqPatternSet;

End

(그림 2) STMP/MST 알고리즘

를 통해 이동 시퀀스 데이터의 공간 속성과 시간 속성을 시공간 질의 분석을 통해 결정된 Ls_k 와 Lt_k 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 트리의 구성을 통해 빈발 이동 패턴을 추출한다.

4.1 이동 시퀀스 생성 알고리즘 : *SeqExtractor*

이동 시퀀스를 생성하기 위한 *SeqExtractor* 알고리즘은 객체 식별자와 객체의 시간 속성인 유효시간을 기준으로 정렬된 이동 객체의 이력 리스트 l 에 대해 1의 두 시간값의 차가 max_gap 보다 크면 리스트를 분리한다. 따라서 단위 시퀀스 변수 $UnitS$ 에 1의 두 시간값의 차가 max_gap 보다 작

을 때까지 1의 값을 $UnitS$ 에 저장하여 시퀀스를 생성한다.

4.2 빈발 이동 패턴 탐사 알고리즘 : *FreqPatternExtractor*

FreqPatternExtractor 알고리즘은 일반화-요약 이동 시퀀스 집합 D_{Tgen} 과 사용자에게 의해 지정된 최소지지도 min_sup 를 전달받아 빈발 이동 패턴을 추출하기 위한 알고리즘이다. *FreqPatternExtractor* 알고리즘에서는 먼저, 일반화-요약 이동 시퀀스 집합으로부터 하나씩 이동 시

Input : D_{rt} (*History Data of Moving Object*), max_gap (*Time Interval Constraint*)

Output : D_{seq}

Procedure SeqExtractor (D_{rt}, max_gap)

Begin

$UnitS = null;$ // $UnitS$: *Unit Sequence*

For each Moving Data List $l \subset D_{rt}$

For ($i = 2; i < l.length; i++$)

$UnitS = l[i - 1];$

If ($(l.vt[i] - l.vt[i - 1]) \geq max_gap$)

$D_{seq} = UnitS;$ // D_{seq} : *Transaction Database*

$UnitS = null;$

End If

End For

End For

Return $D_{seq};$

End

(그림 3) *SeqExtractor* 알고리즘

퀵스를 읽어들이며 해당 이동 시퀀스에 대한 부분 시퀀스 집합을 추출한 후 각 부분 시퀀스에 대한 이동 시퀀스 트리 *MSTrees*를 생성한다. 이동 시퀀스 집합의 모든 이동 시퀀스에 대한 부분 시퀀스 집합 생성 후 각 부분 시퀀스에 대한 이동 시퀀스 트리의 생성이 완료되면, 트리를 구성하는 각 노드의 지지도와 최소지지도 *min_sup*을 비교하여 *min_sup* 보다 작은 지지도를 갖는 노드들을 제거하여 빈발 이동 시퀀스 트리 *FreqMSTrees*를 생성한다. 마지막 빈발 이동 시퀀스 트리를 탐색하여 빈발 이동 시퀀스 집합을 추출한다.

지금까지 제시한 시공간 빈발 이동 패턴 탐사 방법은 다음과 같은 특징을 갖는다.

첫째, 빈발 1-시퀀스 항목들로 구성된 이동 시퀀스 집합에 대한 이동 시퀀스 트리를 구성하고 사용자가 지정한 최소지지도를 만족하는 노드들만으로 구성된 빈발 시퀀스 트리를 생성함으로써 트리 탐색을 통해 잠재적인 빈발 이동 패턴을 쉽게 탐사할 수 있다. 이는 기존의 Apriori 계열 알고리즘들이 빈발 패턴을 탐사하기 위하여, 빈발 1-시퀀스 집합으로부터 각 시퀀스들을 확장해 각 패스마다 후보 시퀀스 집합을 생성하고 데이터베이스를 스캔하여 지지도 계산을 통해 빈발 시퀀

Input : $D_{T_{gen}}$ (*Generalized Moving Sequence*), min_sup (*Minimum Support*)

Output : *OptPath*

Procedure FreqPatternExtractor ($D_{T_{gen}}$, min_sup)

Begin

For each *moving_sequence* $S \in D_{T_{gen}}$

 // *Generating sub_sequence set of moving sequence S*

subSet = *Search Sub_Sequence Set of S*;

 For each *sub_sequence* $S' \in subSet$

 // *Generating moving sequence Trees*

MSTrees ← *Generate hash tree of sub sequence S'*;

 End For

End For

 // *Generating frequent moving sequence trees*

FreqMSTrees ← *delete Node with support smaller than min_sup from MSTrees*;

 // *Generating frequent sequence set*

FreqPattern.Set ← *Search Frequent Sequences from FreqMSTrees*;

 Return *FreqPattern.Set*;

End

(그림 4) FreqPatternExtractor 알고리즘

스 집합을 생성하는 과정을 거치기 때문에 많은 탐사 시간이 소요되는 문제를 해결한다. 즉, 제안된 탐사 방법에서는 이동 시퀀스 트리의 구성과 동시에 각 노드에 시퀀스 지지도를 저장함으로써 빈발 패턴을 보다 효율적으로 탐사할 수 있다.

둘째, Apriori 계열의 알고리즘들은 특정 항목집합을 구성하는 모든 부분집합의 항목들이 모두 빈발할 때 해당 항목집합도 빈발하다는 Apriori 휴리스틱 개념을 근거로 하여 빈발 1-시퀀스 집합의 항목들로 구성되는 이동 시퀀스 집합에 대해 시퀀스 트리를 생성하기 때문에 데이터베이스 스캔을 통해 슈퍼패턴을 탐색하는 것과 동일한 형태의 모든 이동 패턴들을 표현할 수 있다. 마지막으로 제안된 빈발 패턴 탐사 방법에서는 실세계의 공간 및 시간 개념으로 일반화된 이동 시퀀스 집합을 대상으로 패턴 탐사를 수행하기 때문에 탐사되는 패턴들을 지식화하는데 있어 불필요한 데이터 변환이나 가공을 필요로 하지 않는 특성이 있다.

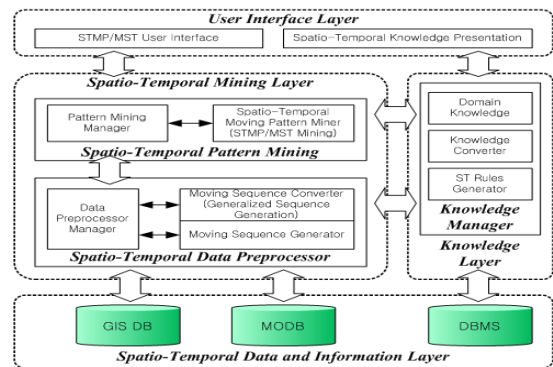
5. 시공간 이동 패턴 탐사 시스템

본 장에서는 앞서 제안한 패턴 탐사 기법을 적용하여 시공간 빈발 패턴을 추출함으로써 시공간 규칙을 생성하기 위한 시공간 이동 패턴 탐사 시스템인 STMPMiner를 설계 및 구현한다. 또한 시스템을 구성하는 각 모듈의 기능 및 시스템 동작 과정을 제시한다.

5.1 시스템 구조

STMPMiner는 이동 객체의 다양한 패턴들 중 주기적이고 순차적인 빈발한 이동 패턴을 탐사하기 위한 시공간 이동 패턴 탐사 시스템으로, 빈발 패턴을 탐사함에 있어 기존의 탐사 기법들이 빈번하게 데이터베이스를 스캔하는 문제를 해결하기 위해서 해시 트리를 이용하여 시간 및 공간 개념 계층 기반의 이동 시퀀스 트리를 생성함으로써 패턴 탐사 시간을 최소화할 수 있도록 지원

다. 또한, 상세한 수준으로 표현되는 이동 객체의 이력 데이터를 패턴 탐사의 배경지식으로서 시간 및 공간 개념 계층을 생성하여 이를 통해 이력 데이터를 일반화함으로써 보다 의미있는 패턴들을 효과적으로 탐사할 수 있도록 유도한다. 다음 (그림 5)는 이러한 기능을 지원하는 시공간 이동 패턴 탐사 시스템인 STMPMiner의 전체 구조이다.



(그림 5) STMPMiner 시스템의 전체 구조

STMPMiner 시스템의 구성은, (그림 5)와 같이 크게 4개의 계층인 사용자 인터페이스 계층(User Interface Layer)과 시공간 마이닝 계층(Spatio-Temporal Mining Layer), 지식 계층(Knowledge Layer), 마지막으로 공간 데이터 & 정보 계층(Spatio-Temporal Data & Information Layer)으로 구성된다. 각 계층의 기능은 다음과 같다.

(표 2) 시스템 구성 계층의 세부 모듈 기능

구성 계층	세부 모듈	기능
사용자 인터페이스 계층	STMP/MST 사용자 인터페이스	- 패턴 탐사 대상이 되는 이동 데이터 집합을 설정하고 부가적인 속성들을 정의하는 인터페이스 제공 - 대상 데이터 집합에 대해 시간 한정과 공간 한정에 대한 표현을 정의하는 인터페이스 제공
	시공간 지식 표현	- 데이터 일반화 시 참조되는 시간 및 공간 개념 계층에 대한 배경지식을 관리하기 위한 인터페이스 제공 - 패턴 탐사를 통해 생성된 지식을 다양한 사용자 요구에 맞게 가공하여 보여주기 위한 지식의 다양한 변환 조건 등을 설정하는 인터페이스 제공
시공간 마이닝 계층	시공간 패턴 탐사	- 시공간 이동 패턴 탐사기 : 순차적이고 주기적인 빈발 이동 패턴을 탐사하는 모듈 - 패턴 탐사 관리자 : 시공간 이동 패턴 탐사기를 제어 및 관리하는 모듈
	시공간 데이터 전처리	- 이동 시퀀스 생성기 : 객체 이력 데이터의 시간 속성에 대해 시간 간격 임계치인 max_gap 을 적용하여 이력 데이터를 분할하여 시퀀스를 생성하는 모듈 - 이동 시퀀스 변환기 : 이동 시퀀스를 공간 및 시간 개념 계층을 이용하여 일반화하는 모듈 - 데이터 전처리 관리자 : 이동 시퀀스 생성기/변환기 모듈을 제어 및 관리하는 모듈
지식 계층	지식 관리자	- 도메인 지식 : 데이터 일반화 시 참조되는 시간 및 공간 개념 계층에 대한 배경지식을 관리하는 모듈 - 지식 표현 : 패턴 탐사를 통해 생성된 지식을 다양한 사용자 요구에 맞게 가공하여 제공해주는 모듈 - 시공간 규칙 생성기 : 시공간 패턴 탐사 모듈에서 전달되는 빈발 이동 패턴들에 대한 시공간 규칙을 생성하는 모듈
시공간 데이터 & 정보 계층	데이터베이스	- GISDB : 이동 객체가 이동 가능한 공간 정보인 지도 데이터를 관리하는 GIS 데이터베이스 - MODB : 이동 객체의 위치 이력 데이터 집합을 관리하는 이동 객체 데이터베이스 - DBMS : 패턴 탐사에 의해 추출되는 시공간 규칙들, 즉 지식을 저장하기 위한 데이터베이스

5.2 시스템 동작 과정

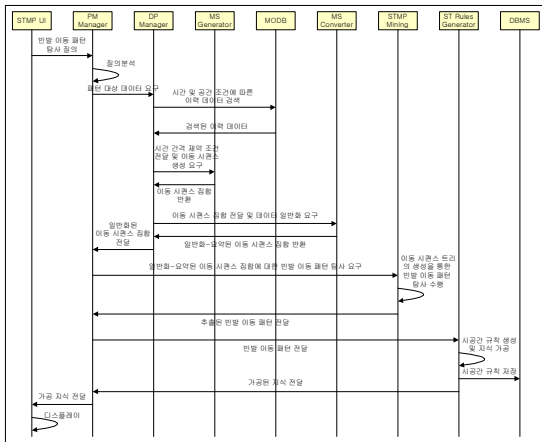
빈발 이동 패턴 탐사를 수행하는 STMPMiner를 구성하는 각 모듈들의 동작과정은 다음과 같다.

- [1] 사용자가 STMP 인터페이스를 통해 순차적이거나 주기적인 빈발 이동 패턴을 탐사하기 위한 질의를 요구한다.
- [2] STMP 인터페이스로부터 전달받은 사용자 요구를 패턴 탐사 관리자는 사용자 질의 분석을 통해 시간 및 공간 제약에 대한 조건들을 데이터 전처리 관리자에게 전달하여 탐사 대상 데이터를 요청한다.
- [3] 데이터 전처리 관리자는 전달받은 시간 및 공간 제약 조건에 적합한 이력 데이터 집합을 생성하기 위해서 이동 객체 데이터베이스를 검색하여 결과를 획득한다.

- [4] 데이터 전처리 관리자는 검색된 이동 객체의 이력 데이터 집합을 이동 시퀀스로 변형하기 위해 시퀀스 생성기에 시간 간격 제약 조건을 전달하여 이동 시퀀스 집합을 생성하도록 지원한다.
- [5] 이동 시퀀스 생성기는 시퀀스 생성 전 먼저 이력 데이터 집합을 이동객체 식별자와 유효시간 순으로 정렬한 후 전달받은 시간 간격 제약 조건을 이용하여 객체의 이력들이 가진 유효시간 간격이 시간 간격 제약을 만족하는지 검사하여 이동 시퀀스를 생성한다.
- [6] 데이터 전처리 관리자는 생성된 이동 시퀀스 데이터를 이동 시퀀스 변환기에 전달하여 공간 및 시간 일반화를 통해 일반화-요약된 시퀀스 집합을 추출한 후 이를 패턴 탐사 관리자에게 전달한다.
- [7] 패턴 탐사 관리자는 전달받은 일반화-요약된 이동 시퀀스 집합으로부터 시공간 패턴 탐사 모듈을 통해 이동 시퀀스 트리를 생성하여 주기적이거나 순차적인 빈발 이동 패턴을 추출한다.

- [8] 추출된 빈발 이동 패턴은 시공간 규칙 생성기에 전달하여 이동 패턴에 대한 시공간 규칙을 생성한 후 데이터베이스에 저장한다. 또한 시공간 규칙 생성기는 지식 변환기를 이용하여 생성된 규칙을 가공하여 패턴 탐사 관리자에게 전달한다.
- [9] 패턴 탐사 관리자는 이를 STMP 인터페이스에 전달하여 사용자에게 지식을 제공하도록 한다.

다음 (그림 6)은 이러한 빈발 패턴 탐사 시 STMPMiner의 각 모듈 간 동작과정을 시퀀스 다이어그램(sequence diagram)으로 표현한 것이다.



(그림 6) 빈발 패턴 탐사 시 시스템 동작 과정

6. 실험 및 성능 평가

실험을 위한 시공간 이동 패턴 탐사 시스템인 STMPMiner의 개발을 위해 Windows XP 기반에서 개발도구 Eclipse SDK 3.2.2와 JDK 1.6.0을 사용하였고, 이동 객체 데이터와 지형(geometry) 데이터를 저장하기 위해서 Oracle 10g로 데이터베이스 시스템을 구축하였다. 빈발 패턴 탐사 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, 이

동 객체의 이력 데이터로 서울시의 도로 네트워크 상에서 택시들의 운행 기록을 추출하여 이력 데이터를 생성하였다. 추출된 이력 데이터는 도로 네트워크 상에서의 교차점이나 분기점, 종료점 등을 노드로 설정하여 각 노드에서 이동 객체의 위치 정보를 샘플링하였다. 다음 (표 2)는 성능 평가 실험에 이용되는 이력 데이터의 유형이고, (표 3)은 실험 데이터에 대한 특성을 나타낸다. 데이터 집합의 이름에서 N은 이동 객체 수, D는 이동기간, G는 샘플링 위치를 의미한다.

(표 3) 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 4) 실험 데이터 특성

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

제한된 STMP/MST 알고리즘과 비교 및 분석을 통해 성능을 평가할 기존의 패턴 탐사 기법으로 MPMine과 Apriori-MSP 알고리즘을 대상으로 선택하였다. 실험을 통해 각 알고리즘과 비교할 평가 기준은 최소지지도, 시간 한정 크기, 이동 객체 수 등의 변수를 사용하여 패턴 탐사 수행 시간에 대한 효율성을 측정한다. 실험 방식은 세 개의 변수 중 두 가지 변수를 고정하고 나머지 한 가지 변수의 값을 변화시키면서 실험을 수행하였다.

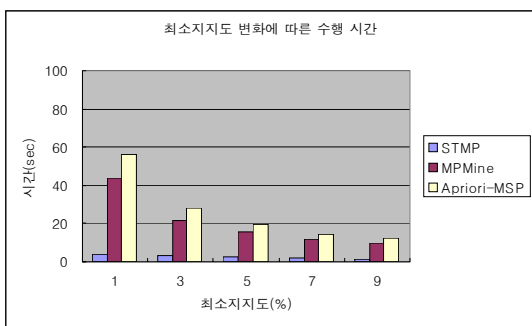
5.1 최소지지도 변화에 따른 수행 시간 평가

(표 4)는 이동 객체 수를 1000개로 고정하고 시간 한정 크기는 5시간으로 고정하여 최소지지도를 1%, 3%, 5%, 7%, 9%로 변화시키면서 수행 시

간을 측정 한 결과이다. (그림 7)에서 STMP/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 MPMine과 Apriori-MSP에서 패턴 탐사를 위해 생성하는 후보 시퀀스의 수가 증가하면서 잦은 데이터베이스 스캔을 필요로 하고, 각 후보 시퀀스와의 비교 연산 시간이 증가하여 큰 폭으로 수행 시간이 증가하기 때문이다. 하지만 STMP/MST의 경우 최소지지도의 변화에 있어 일정한 탐사 시간을 유지하는 이유는 후보 시퀀스를 생성하지 않아 데이터베이스 접근이 불필요함으로써 상대적으로 훨씬 탐사 수행 시간이 적게 소요된다.

(표 5) 최소지지도 변화에 따른 수행 시간 결과

최소지지도	STMP/MST	MPMine	Apriori-MSP
1	3.72	43.47	56.38
3	3.16	21.34	27.97
5	2.56	15.68	19.81
7	1.84	11.94	14.35
9	1.43	9.87	12.37



(그림 7) 최소지지도에 변화에 따른 수행 시간 비교

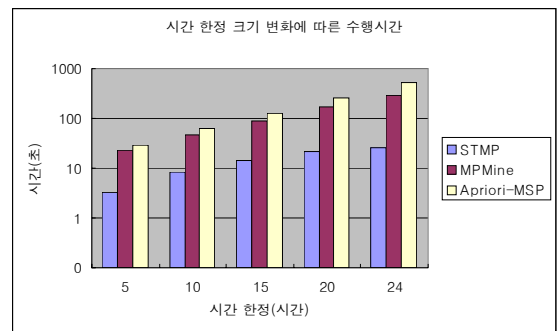
5.2 시간 한정 크기 변화에 따른 수행 시간 평가

(표 5)는 이동 객체의 수 1000개, 최소지지도를 5%로 고정하고 시간 한정 크기를 5시간, 10시간,

15시간, 20시간, 24시간으로 변화시키면서 탐사 수행 시간을 측정 한 결과이다. (그림 8)을 보면 STMP/MST가 MPMine과 Apriori-MSP보다 상대적으로 적은 증가율을 보이며 패턴 탐사 수행 시간도 훨씬 적게 걸린다. 이는 STMP/MST가 패턴 탐사 시 데이터 I/O를 단 두 번만 수행하여 I/O 시간을 최소화하기 때문에 더 좋은 성능을 보인다. 또한, 이동 시퀀스 트리를 구성하여 후보 시퀀스를 생성하지 않고 빈발 패턴을 탐사하기 때문에 후보 시퀀스의 검증에 대한 데이터베이스 접근이 불필요하여 탐사 시간을 효율적으로 감소시킬 수 있다.

(표 6) 시간 한정 크기 변화에 따른 수행 시간 결과

시간 한정 크기	STMP/MST	MPMine	Apriori-MSP
5	3.24	22.47	29.34
10	8.47	47.67	62.38
15	14.37	89.41	128.47
20	20.98	172.34	255.13
24	26.23	297.55	511.85



(그림 8) 시간 한정 크기 변화에 따른 수행 시간 비교

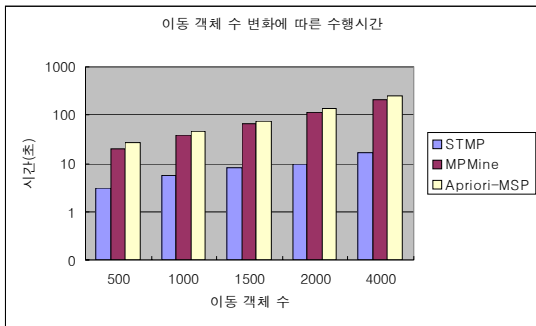
5.3 이동 객체 수 변화에 따른 수행 시간 평가

(표 6)은 최소지지도 5%, 시간 한정 크기를 5시간으로 고정하고 이동 객체의 수를 500개, 1000개, 1500개, 2000개, 4000 개로 변화시키면서 탐사 수행 시간을 측정 한 결과이다. (그림 9)에서 보는 바와 같이 이동 객체 수가 증가함에 따라 모든 알

고리즘의 수행 시간도 점차적으로 증가하고 있다. 이는 이동 객체의 수가 증가함에 따라 데이터양도 많아져 후보 시퀀스에 대한 비교 연산의 증가로 탐사 수행시간이 증가하는 것이다. 하지만 STMP/MST가 다른 알고리즘에 비해 전반적으로 빠른 수행시간을 보인다. 이는 앞서 언급한 바와 같이 이동 시퀀스 트리를 구성하여 지지도를 계산하기 때문에 후보 시퀀스의 검증을 위한 데이터베이스 스캔이 불필요함으로써 탐사 수행 시간이 적게 소요된다.

(표 7) 이동 객체 수 변화에 따른 수행 시간 결과

이동 객체 수	STMP/MST	MPMine	Apriori-MSP
500	3.12	20.45	25.64
1000	5.64	38.64	47.34
1500	7.89	65.17	76.55
2000	9.87	115.53	139.42
4000	16.25	206.86	248.64



(그림 9) 이동 객체 수 변화에 따른 수행 시간 비교

6. 결론 및 향후 연구과제

시공간 이동 패턴 탐사는 연속되는 시간상에서 발생하는 객체의 다양한 패턴들 중 독특한 이동 경향을 갖는 패턴이나 반복적이고 공통적인 패턴을 탐사하여 새로운 위치 기반 서비스를 개발하기 위한 기술이다. 기존의 패턴 탐사 방법들은 빈발 패턴 탐사 시 빈번하게 데이터베이스를 접근

함으로써 탐사 수행 시간이 증가하는 문제와 탐사 대상 데이터의 양에 따라 탐사 시 소요되는 메모리량이 증가하여 탐사 성능이 저하되는 문제가 있다. 이러한 기존의 탐사 기법에 대한 문제점을 해결하기 위하여 본 논문에 앞선 선행 연구로 [11]에서는 방대한 시공간 이동 데이터 집합으로부터 순차적이고 주기적인 빈발한 이동 패턴을 탐사하는데 있어 탐사 수행 시간 및 탐사 시 소요되는 메모리량을 최소화하기 위해서 시공간 데이터 일반화와 이동 시퀀스 트리 기반의 패턴 탐사 방식을 통해 보다 효과적으로 패턴을 탐사하기 위한 STMP/MST 탐사 기법을 제시하였다. 제안된 기법은 이동 시퀀스 트리를 기반으로 패턴 탐사를 수행함으로써 데이터베이스 접근을 최소화하여 탐사 수행 시간을 감소시키고, 상세 수준의 이력 데이터를 실제계의 공간 및 시간 개념으로 일반화하여 탐사 시 소요되는 메모리량을 최소화할 수 있다. 본 논문에서는 제안된 STMP/MST 탐사 기법의 탐사 수행 시간에 대한 효율성을 검증하기 위해서 기존의 패턴 탐사 기법인 MPMine[9]과 Apriori-MSP[10] 알고리즘을 대상으로 빈발 패턴 탐사 시의 최소지지도와 데이터의 시간 한정 크기, 이동 객체 수의 변화에 따른 패턴 탐사 수행 시간에 대한 성능을 비교하고 분석하였다.

실험 결과 각각의 평가 기준 변화 시 제안된 STMP/MST 알고리즘이 MPMine과 Apriori-MSP 알고리즘에 비해 훨씬 좋은 성능을 보였다. 이는 STMP/MST 알고리즘이 이동 시퀀스 트리를 구성하여 패턴 탐사를 수행하기 때문에 후보 시퀀스를 생성하지 않음으로써 이에 대한 검증 및 별도의 지지도 계산에 대한 연산을 필요로 하지 않고, 데이터베이스 접근을 단 2회로 최소화하여 전반적인 탐사 수행 시간을 감소시킴으로써 효과적인 패턴 탐사를 수행할 수 있었다. 성능 평가를 통해 알 수 있듯 제안된 기법은 기존의 탐사 기법들에 비해 효율적인 패턴 탐사를 지원함으로써 다양한 응용 분야에 활용할 수 있는 새로운 위치 기반 서비스를 개발할 수 있다. 위치 기반 서비스로의 활용은 사용자의 이동 경로나 프로필 등의 성향을

이용하여 유용한 지식을 추출함으로써 각 사용자의 동적인 위치에 따라 특화되고 개인화된 위치 기반 서비스를 제공할 수 있다. 가령, 각종 기업들은 사용자의 이동 경로를 예측하여 제품이나 기타 상품, 또는 공익에 대한 광고를 보여줄 수 있으며, 사용자는 현재 자신이 위치한 지역을 중심으로 비교 쇼핑과 같은 상거래 서비스를 제공할 수 있다.

참 고 문 헌

- [1] D. O. Kim, H. K. Kang, D. S. Hong, J. K. Yun and K. J. Han, "STMP : 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] 백옥현, "위치 기반 서비스를 위한 이동 객체의 시간 패턴 탐사 기법", 충북대학교 대학원, 석사학위논문, 2002.
- [8] 이준욱, "위치 기반 서비스를 위한 이동 객체의 시간 패턴 탐사", 한국정보과학회 논문지, 제29권, 제5호, 2002.
- [9] 이준욱, "지식 탐사 프레임워크 기반의 시공간 이동 패턴 탐사 기법", 충북대학교 대학원, 박사학위논문, 2003.
- [10] 한선영, "시공간 이동 시퀀스 패턴 마이닝 기법", 이화여자대학교 대학원, 석사학위논문, 2006.
- [11] 고 현, 이연식, "이동 시퀀스 트리를 이용한 효율적인 시공간 이동 패턴 탐사 기법", 한국정보처리학회논문지(심사완료).
- [12] 안병익, "LBS 기술 동향과 전망: LBS 구조 및 구성", 한국지리정보 통권 56호, pp.52-56. 2001. 10.
- [13] 안병익, "LBS 기술 동향과 전망: LBS 소프트웨어 기술 동향", 한국지리정보 통권 57호, pp.68-72. 2001.11.
- [14] 안병익, "LBS 기술 동향과 전망: LBS 서비스 내용", 한국지리정보 통권 58호, pp.70-75. 2001.12.
- [15] 양영규, "위치기반 서비스(LBS:Location-Based Service) 기술 현황 및 전망", 한국정보처리학회 학회지, 제8권, 제6호, 2001.
- [16] 박지웅, "시공간 이동 패턴 추출을 위한 효율적인 알고리즘", 건국대학교 대학원, 박사학위논문, 2006.
- [17] 고 현, 김광중, 이연식, "최적 경로 서비스 지원을 위한 이동 객체의 이동 패턴 탐사 알고리즘", 한국정보처리학회 추계학술발표논문집, 제7권, 제2호, 2006. 11.
- [18] 고 현, 김광중, 이연식, "이동 객체의 패턴 마이닝을 위한 위치 일반화 방법", 한국정보처리학회 추계학술발표논문집, 제7권 제2호, 2006. 11.
- [19] 고현, 김광중, 이연식, "이동 객체의 패턴 탐사를 위한 시공간 데이터 일반화 기법", 한국정보과학회 한국컴퓨터종합학술대회논문집(C), 제34권, 제1호, 2007. 06.

- [20] 고현, 김광중, 이연식, “R*-Tree와 Grid를 이용한 이동 객체의 위치 일반화 기법”, 한국컴퓨터정보학회논문지, 제12권, 제2호, 2007. 05.

● 저 자 소 개 ●



이 연 식

1982년 전남대학교 전자계산학과(학사)
1984년 전남대학교 대학원 전자계산학과(이학석사)
1994년 전북대학교 대학원 전산응용공학전공(공학박사)
1995년~1997년 군산대학교 교무부처장
1997년~1998년 University of Missouri 교환교수
1999년~2001년 군산대학교 전자계산소 소장
2004년~2005년 Ohio State University 교환교수
1986년~현재 군산대학교 컴퓨터정보공학과 교수
관심분야 : 번역기 이론, 객체지향시스템, 능동시스템, 지능형 에이전트, 센서 네트워크 미들웨어
e-mail : yslee@kunsan.ac.kr



김 은 아

1990년 광운대학교 전자계산학과(학사)
1998년 광운대학교 전산대학원 전자계산학과(이학석사)
2000년 ~ 현재 한국폴리텍 II인천캠퍼스 컴퓨터정보과 교수
관심분야 : 객체지향 시스템, 데이터 마이닝, 지능형 에이전트
e-mail : eakim@kopo.ac.kr