

# 이동 객체의 패턴 탐사를 위한 시공간 데이터 일반화 기법

고현, 김광종, 이연식  
 군산대학교

[khyun001@kunsan.ac.kr](mailto:khyun001@kunsan.ac.kr), [kkjkim@kunsan.ac.kr](mailto:kkjkim@kunsan.ac.kr), [yslee@kunsan.ac.kr](mailto:yslee@kunsan.ac.kr)

## KISS Korea Computer Congress 2007

Hyun Ko, Kwangjong Kim, Yon-sik Lee

### 요 약

사용자들의 특성에 맞게 개인화되고 세분화된 위치 기반 서비스를 제공하기 위해서는 방대한 이동 객체의 위치 이력 데이터 집합으로부터 유용한 패턴을 추출하여 의미 있는 지식을 탐사하기 위한 시공간 패턴 탐사가 필요하다. 현재까지 다양한 패턴 탐사 기법들이 제안되었으나 이동 패턴들 중 단순히 시공간 제약이 없는 빈발 패턴만을 추출하기 때문에 한정된 시간 범위와 제한적인 영역 범위 내에서의 빈발 패턴을 탐사하는 문제에는 적용하기 어렵다. 또한 패턴 탐사 수행 시 데이터베이스를 반복 스캔하여 탐사 수행 시간이 많이 소요되는 문제를 포함하거나 메모리상에 탐사 대상인 후보 패턴 트리를 생성하는 방법을 통해 탐사 시간을 줄일 수는 있으나 이동 객체 수나 최소지지도 등에 따라 트리를 구성하고 유지하는데 드는 비용이 커질 수 있다. 따라서 이러한 문제를 해결하기 위한 효율적인 패턴 탐사 기법의 개발이 요구됨으로써 선행 작업으로 본 논문에서는 상세 수준의 객체 이력 데이터들의 시간 및 공간 속성을 의미 있는 시간영역과 공간영역 정보로 변환하는 시공간 데이터 일반화 방법을 제안한다. 제안된 방법은 공간 개념 계층에 대한 영역 정보들을 영역 Grid 해쉬 테이블(AGHT:Area Grid Hash Table)로 생성하여 공간 인덱스 트리인 R\*-Tree의 검색 방법을 이용해 이동 객체의 위치 속성을 2차원 공간영역으로 일반화하고, 시간 개념 계층을 생성하여 이동 객체의 시간적인 속성을 시간 영역으로 일반화함으로써 일반화된 데이터 집합을 형성하여 효율적인 이동 객체의 시간 패턴 마이닝을 유도할 수 있다.

### 1. 서 론

교통관제, 텔레매틱스, 기상예측, 마케팅 등 다양한 응용 분야에 활용될 수 있는 서비스를 개발하기 위해서는 시간, 공간 및 시공간 특성을 포함하고 있는 방대한 데이터 집합으로부터 이전에 알려지지 않았던 잠재적으로 유용한 시공간 지식을 탐사하기 위한 시공간 데이터 마이닝이 필요하다. 기존의 시공간 데이터 마이닝에 대한 연구들 중 시공간 패턴 탐사 기법[3,4,5,6,7,8,13,14,15]은 시간 순서나 변화에 따라 규칙적이고 반복적으로 발생하는 이벤트의 패턴을 탐사하기 위한 기법으로 시계열 데이터 및 시퀀스 데이터 분석에 이용될 수 있다. 하지만 시공간 패턴 탐사 기법은 지식 탐사를 위해서 이동 객체의 이력 데이터에 대한 일반적인 특성을 분석하여 사용하기 알맞은 형태로의 데이터 변환을 필요로 한다. 이는 이동객체의 연속적인 이동 위치가 이산적인 시점에서 샘플링 된다고 가정할 때 평면상의 (x, y)좌표 값 형태로 표현되어 의미 있는 패턴을 찾기가 매우 어렵기 때문이다. 따라서 기존의 탐사 기법들은 시공간 속성을 고려하여 이동 객체의 위치 및 각 위치에서의 유효시간을 사용자 정의에 따라 일반화된 값으로 변환하는 시공간 일반화 연산을 정의하여 사용한다. 기존의 패턴 탐사에서 사용하는 일반화 방법들은 단순히 공간영역이나 시간영역에 대해 개별적으로 일반화 방법만을 적용하거나, 두 영역 모두에 대해 일반화 방법을 사용하지만 실제세계의 공간영역과 지식의 응용 분야별 시간계층에 대한 고려없이

동일한 크기의 셀 영역으로 분할하여 일반화를 수행하기 때문에 의미 있는 실제세계의 공간영역과의 매핑이 필요하다. 또한, 이동 객체의 위치나 시간 속성을 단순히 하나의 공간 영역 또는 시간 영역으로만 일반화하기 때문에 공간 영역이나 시간 영역에 존재할 수 있는 각 영역들 간의 포함 관계에 따른 계층 구조를 고려한 레벨 수준별 공간 및 시간 계층에 대한 정의가 필요하고 이를 기반으로 일반화 방법의 적용이 요구된다.

이에 본 논문에서는 상세 수준의 이동 객체 데이터를 일반화하기 위해서 이동 객체의 위치 값과 공간영역 간의 위상관계를 고려하여 특정 공간 관념에 따른 레벨별 공간 개념 계층을 정의한다. 또한 지식 응용 분야에 대한 시간적인 관념을 고려하여 특정 시간의 의미를 갖는 시간영역들로 구성된 시간 개념 계층을 정의하여 이를 기반으로 일반화하는 방법을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구로 기존의 이동 객체 패턴 탐사 기법과 일반화 방법에 대해 기술하고, 3장에서는 공간 영역과 시간 영역에 대한 개념 계층을 정의한다. 4장에서는 이동 객체 데이터의 위치 속성과 시간 속성을 일반화하기 위한 공간 및 시간 연산 알고리즘을 제안하고, 5장에서는 결론 및 향후 연구 과제를 제시한다.

### 2. 관련연구

시공간 데이터 마이닝은 시공간 특성을 포함하고 있는

방대한 시공간 데이터 집합으로부터 묵시적이고 잠재적으로 유용한 지식을 추출하는 기술로 정의될 수 있으며, 시공간 객체들의 공통적이고 일반적인 성질을 이용하여 객체들을 집단화하는 기술, 시공간 객체들 간의 시간, 공간 및 시공간 상에서의 연관 관계, 시공간 상에서의 변화에 대한 패턴 및 예측과 같은 것을 목적으로 한다[14]. 현재까지 시공간 데이터 마이닝에 대한 많은 연구가 수행되었으며, 기존의 연구들은 접근 방법에 따라 높은 유사도를 가지는 객체들을 같은 클러스터로 그룹화하는 시공간 클러스터링 방법[1,2]과 시공간 상에서 객체의 이동 변화에 대한 패턴을 탐사하는 시공간 패턴 탐사 방법[3,4,5,6,7,8, 13,14,15]으로 구분될 수 있다. 클러스터링 기반의 시공간 데이터 분석은 시공간 객체들의 공통적이고 일반적인 성질을 이용하여 객체들을 집단화하는 과정으로 패턴 탐사와 같은 다른 마이닝 기법의 전처리 단계로 이용되거나 유사성 검색 등의 응용 분야에 적용될 수 있다. 반면, 시간 순서에 따라 발생하는 이벤트나 시간 변화에 따라 규칙적이고 반복적으로 발생하는 이벤트의 패턴을 탐사하기 위한 패턴 탐사 기법은 시계열 데이터 및 시퀀스 데이터 분석에 이용될 수 있다.

한편, 데이터 마이닝 기법들 중 기존의 일반적인 데이터에 존재하는 패턴을 탐사하기 위한 기법들로 순차 패턴[9,10,11], 주기 패턴[12] 등이 있다. 이러한 기법들은 시간 및 공간 속성, 시공간 속성에 대한 고려를 통해 시공간 데이터의 패턴 추출을 위해 개발된 여러 기법들의 기반이 되었다. 현재까지 연구된 시공간 패턴 탐사 기법들로는 STPMine1[3], STPMine2[3], GSP[4], MP[5,6], MPMine[13,14], STMPE[15] 등이 있다. 먼저, Apriori 계열의 알고리즘을 변형한 STPMine1[3] 기법은 이동 객체의 이력 데이터로부터 주기적 패턴을 추출하는 기법으로, Apriori 계열 알고리즘이 패턴 길이에 따라 데이터베이스를 반복 스캔하는 문제를 그대로 포함하고 있다. 이러한 문제로 STP-Mine2[3]에서는 빈발 1-패턴을 조합한 Super와 Shadow 패턴을 생성, Max\_Subpattern 트리를 구성하여 빈발 패턴을 추출함으로써 데이터베이스의 스캔 횟수를 크게 줄이는 방법을 사용하였다. 그러나 이 또한, Super와 Shadow 패턴의 최소지지도가 낮거나 또는 이동 객체 수가 증가하거나, 시간 영역의 수가 늘어날수록 후보 패턴의 수가 증가하여 비교 연산 횟수 및 소요 메모리의 양이 증가하는 단점이 있다. GSP[4]도 Apriori 알고리즘에 기반한 순차 패턴 추출 알고리즘으로, 시간 제한(Time Constraints), 이동 윈도우(Sliding Time Windows), 분류(Taxonomies) 등을 도입하여 순차 패턴 추출을 일반화하지만, 많은 데이터베이스 스캔 시간이 필요하고 긴 순차 패턴의 마이닝 시 성능이 저하되는 단점이 있다. STMPE[15]는 이동 객체의 이력 정보를 분석하여 시공간 정보를 포함하는 일반화된 이동 패턴을 추출하는 알고리즘으로, 시공간 데이터 일반화, 이동 패턴 추출, 이동 패턴 트리 생성 기능을 지원하여 일반화를 통해 시공간 데이터 저장을 위한 메모리 사용량을 최소화하고, 이동 패턴 트리를 생성하여 유지함으로써 데이터베이스 스캔 횟수를 최소화하여 탐사 수행 시간을 감소시킬 수 있지만 이동 패턴 트리의 크기가 커질수록 많은 저장공간을 소모하게 된다. 마지막으로 MP[5,6]와 MPMine[13,14]은

Apriori 알고리즘을 적용한 알고리즘으로, 전처리 과정에서 이동 객체의 이력 데이터 집합으로부터 위치 요약 및 일반화를 통해 탐색공간을 줄일 수 있는 효율적인 패턴 탐사 기법이다.

지금까지 고찰한 시공간 패턴 탐사 기법들은 묵시적 또는 명시적으로 일반화 개념을 적용하여 패턴 탐사를 수행하였다. STMPE[15]와 Apriori\_msp[16]은 이동 객체의 시간 속성과 공간 속성을 각각 시간 및 공간영역으로 일반화하였으나, 실세계의 시간 개념 및 공간 개념에 대한 고려없이 단순히 마이닝 대상 데이터 자체를 사용자가 정의한 특정 셀단위로 분할하여 일반화를 수행하기 때문에 일반화된 데이터를 실세계의 의미있는 영역으로 사상하기 위한 매핑과정이 필요하다. 또한, MP[5,6]와 MPMine[13,14]에서는 일반화하기 위한 공간을 실세계의 공간지역에 대한 추상 개념 계층으로 표현하여 이동 객체의 위치를 일반화하는 방법만을 제안하였을 뿐 시간 속성을 일반화하기 위한 방법에 대한 고려가 제시되지 않았다. 따라서 이동 객체의 연속적인 위치 변화를 보다 효과적으로 패턴화하기 위해서는 기존의 일반화 방법들이 가지고 있는 문제점을 보완한 새로운 시공간 일반화 접근법이 필요하다.

### 3. 시간 개념 계층과 공간 개념 계층

#### 3.1 공간 개념 계층

실세계에서 이동 객체는 끊임없이 공간을 이동하며 무한한 위치 정보를 발생시키기 때문에 (x, y) 형태의 상세화된 위치 정보를 가지고 의미있는 패턴을 탐색하기에는 많은 어려움이 따른다. 따라서 의미있는 지식을 획득하기 위해서 이동 객체의 상세한 위치 정보를 공간 상의 상위 개념으로 추상화할 필요가 있다. 이동 객체의 위치 속성은 공간 상의 특정 지역 수준의 의미로 일반화되기 위하여 공간 지역에 대한 공간 개념 계층이 요구된다.

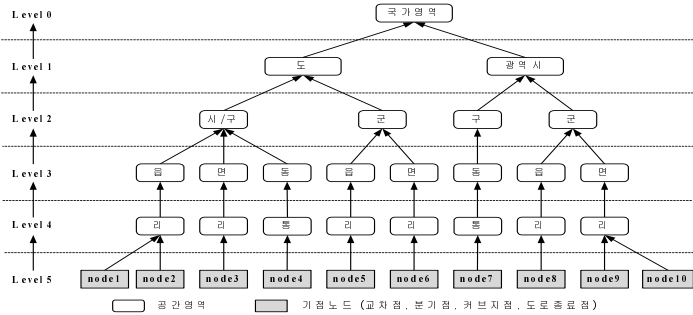
**[정의 1]** 이동 객체가 이동 가능한 지역(Region)들의 집합  $R = \{R_i | R_i \in S, \text{ for } \forall i\}$ 이며, 지역  $R$ 의 인스턴스 객체  $R_i = \langle R_{id}, R_S, R_A \rangle$ 로 표현한다.  $R_{id}$ 는 지역 식별자이며,  $R_S$ 와  $R_A$ 는 지역의 공간속성과 일반속성이다.

**[정의 2]** 공간 개념 계층 레벨의 집합  $SL = \{SL_j\}$ 이고,  $SL_k \leq SL_{k-1}$ 이다. 이 때,  $0 \leq k \leq j$  이다.

**[정의 3]** 공간 개념 계층 레벨  $SL$ 에 대한  $R$ 의 인스턴스 객체는 다음과 같다.

$$SL_{R_i} = \{SL_{k_R} | SL_{k_R} \subset R, SL_{k_R} \in SL_{k-1_R}, 1 \leq k \leq j \text{ and for } \forall i\}$$

이러한 정의들에 의거하여 공간 지역에 대한 개념 계층을 생성할 수 있다. 다음 [그림 1]은 행정구역에 대한 6 레벨의 공간 개념 계층의 예이다. 단말 수준의 노드에서는 도로 상의 교차점, 분기점, 도로종료점 등을 표현하고 각 레벨별 공간 범위는 포함관계를 통해 상위 레벨 영역으로 일반화된다.



[그림 1] 행정구역에 대한 공간 개념 계층

3.2 시간 개념 계층

이동 객체의 이력 데이터는 3.1에서 기술한 바와 같이 상세화된 위치값과 각 위치에서의 유효시간으로 구성되기 때문에 패턴 탐사를 위해서 객체의 위치 속성에 대한 공간 개념 계층으로의 일반화 뿐만 아니라 유효시간도 패턴 분석을 통해 추출되는 지식의 응용 분야에 따라 시간적인 관념을 고려하여 추상적인 개념 계층으로 일반화해야 한다.

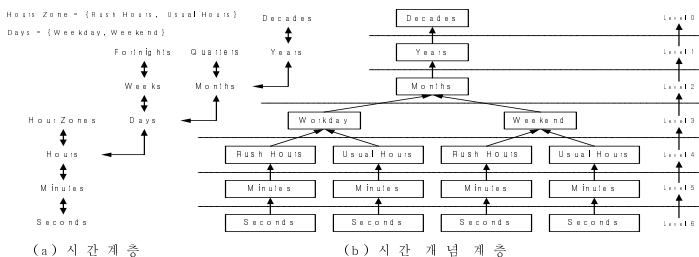
**[정의 4]** 이동 객체가 이동 시 샘플링된 유효시간 (Valid Time)들의 집합  $VT = \{VT_i | VT_i \in T, \text{ for } \forall i\}$ 이며, 유효시간  $VT$ 의 인스턴스 객체  $VT_i = \langle VT_{id}, VT_T, VT_A \rangle$ 로 표현한다.  $VT_{id}$ 는 유효시간 식별자이며,  $VT_T$ 와  $VT_A$ 는 의 시간속성과 일반속성이다.

**[정의 5]** 유효시간 개념 계층 레벨의 집합  $TL = \{TL_j\}$  이고,  $TL_k \leq TL_{k-1}$ 이다. 이 때,  $0 \leq k \leq j$  이다.

**[정의 6]** 시간 개념 계층 레벨  $TL$ 에 대한  $VT$ 의 인스턴스 객체는 다음과 같다.

$$TL_{VT_i} = \{TL_{k_{VT}} | TL_{k_{VT}} \subset VT, TL_{k_{VT}} \in TL_{k-1_{VT}}, 1 \leq k \leq j \text{ and for } \forall i\}$$

이러한 정의들에 의거하여 이동 객체의 시간속성을 일반화 하는데 필요한 시간 개념 계층을 생성할 수 있다. [그림 2-(a)]는 일반적인 시간 계층에 대한 표현이고 [그림 2-(b)]는 Road-Network 상에서 이동 객체의 이동 시간 (유효시간)에 대한 7 레벨의 시간 개념 계층의 예이다.



[그림 2] 객체의 이동시간에 대한 시간 개념 계층

Level 3에서의 Workday(주중)와 Weekend(주말) 시간기간은 Days의 부분집합으로 동일한 영역에 대한 Workday 기간에서의 이동 시간과 Weekend 기간에서의 이동 시간이 다르기 때문에 두 개의 시간 영역으로 분류한다. 또한, Rush Hours와 Usual Hours 시간 간격에서도 서로 다

른 이동 시간 범위를 가질 수 있으므로 Day를 두 개의 시간대로 분류한다.

4. 이동 객체의 시간 및 공간 속성의 일반화

4.1 이동 객체의 위치 일반화

4.1.1 Contains 연산

위치 일반화는 객체의 위치값과 영역 간의 공간 관계에 대한 분석을 통해 이동 객체의 위치를 공간 영역으로 일반화하는 것으로, 공간 객체들의 위상 관계를 고려하여 2차원 공간 객체들 사이에 가능한 관계를 분석한다. OGC(Open GIS Consortium)의 구현에 대한 명세에는 Disjoint, Touches, Crosses, Within, Overlaps의 위상 관계 연산과 Contains, Intersects 라는 부가적인 위상 관계 연산을 정의하고 있다. 이러한 위상 관계 연산들 중 이동 객체의 위치에 대한 공간 영역으로의 일반화를 위해서는 Contains 연산 정의를 이용할 수 있다.

이동 객체가 샘플링된 위치를 공간영역으로 일반화하는 Contains 공간연산은 객체의 위치 좌표점이 공간 개념 계층에서 레벨 수준별 영역의 최소경계사각형 (MBR:Minimum Bounding Rectangle)에 포함되었는지를 검사하고 해당 MBR 영역에 대한 실영역들의 경계에 위치하거나 또는 영역 내부에 포함되었는지를 검사하여 이동 객체의 위치 속성을 공간영역으로 일반화한다. Contains 연산을 위한 선행 작업으로는 시공간 질의를 분석하여 공간 개념 계층의 레벨 수준을 결정해야 한다.

```

Number of Spatial Concept Level: N
Set of Spatial Concept Level: L = {L0, L1, ..., Ln-1}, 0 ≤ n ≤ N-1
Number of Area MBR in Level Li: M
Set of Area MBR in Level Li: Li = {MBR1, MBR2, ..., MBRm}, 1 ≤ m ≤ M
Current Location Coordinates of Moving Object: MP
Previous Area contained MP: PrevA
Current Area contained MP: CurrA

Input : MP, PrevA, N
Output : CurrA

Procedure Contains (MP, PrevA, N)
Begin
    L0 = {MBR1, MBR2, ..., MBRm};
    ContMBR = ∅; // Set of MBRs Contained MP

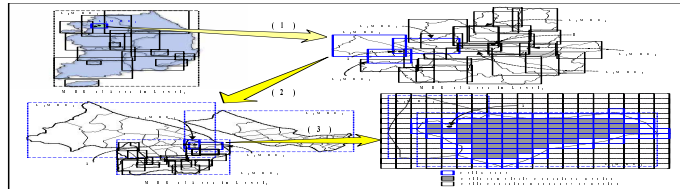
    For (i = 0 to n-1)
        If (ContMBR ≠ ∅) Then
            For (j = 1 to ContMBR.Count) Then
                Li is low level MBRs of ContMBRj;
                LowMBR = ContainedMBR(MP, Li);
                ContMBR = LowMBR;
            End For
        Else
            ContMBR = ContainedMBR (MP, Li);
        End If
    End For

    Return ContainedArea(ContMBR, MP, PrevA);
End
    
```

[그림 3] Contains 공간 연산 알고리즘

[그림 3]의 Contains 연산은 공간 개념 계층의 레벨 수

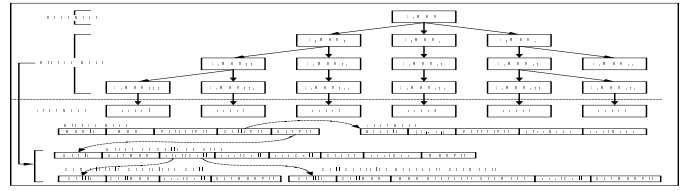
준에 따라 전체의 공간 영역을 시공간 질의 분석을 통해 결정된 공간 계층 레벨 수준(N)까지 공간영역을 반복적으로 MBR로 분할하며 ContainedMBR 연산을 통해 이동 객체의 위치 좌표점(MP)이 어느 레벨 수준의 영역 MBR에 포함되는지를 검사한다. 이 때, 최상위 레벨 수준에서 하위 레벨 수준까지 MBR로 분할하는 이유는 시공간 질의 수준에 맞게 공간 개념 계층의 레벨 수준으로 이동 시퀀스를 생성하기 위함이다. 예를 들면, [그림 4]와 같이 이동 객체가 A지점과 B지점에 있을 경우, Level 0 수준의 영역을 Level 1 수준의 하위 영역(L<sub>1</sub>MBR)으로 분할한 후 각 영역에 대한 L<sub>2</sub>MBR을 생성하고 위치 좌표점 A와 B를 포함하는 L<sub>2</sub>MBR들을 추출한다(1). 위치 좌표점 A는 L<sub>2</sub>MBR<sub>1</sub>, L<sub>2</sub>MBR<sub>2</sub>, L<sub>2</sub>MBR<sub>3</sub>에, B는 L<sub>2</sub>MBR<sub>4</sub>, L<sub>2</sub>MBR<sub>5</sub>, L<sub>2</sub>MBR<sub>6</sub>에 각각 포함된다. 또한, [그림 4]와 같이 좌표점이 포함되어 있는 Level 2의 MBR에 대해 반복적으로 Level 3의 MBR을 생성하여 A, B를 포함한 Level 3 수준의 영역 MBR을 탐색한다(2). A와 B는 각각 L<sub>3</sub>MBR<sub>1</sub>, L<sub>3</sub>MBR<sub>2</sub>와 L<sub>3</sub>MBR<sub>3</sub>에 포함된다.



[그림 4] L<sub>1</sub>MBR→L<sub>2</sub>MBR→L<sub>3</sub>MBR→Grid 표현

4.1.2 ContainedArea 연산

[그림 6]의 ContainedArea 연산은 위치 좌표점을 포함하는 MBR 집합(ContMBR), 이동 객체의 위치 좌표점(MP), 이동 객체의 이전 위치 포함 영역(PrevA)에 대한 정보를 전달받아 각 영역 MBR에 대응하는 실영역에 위치 좌표점이 포함되는지 검사하여 해당 영역의 이름을 반환하는 공간 연산 함수이다. 이 때, 반환되는 영역 MBR의 이름은 실제 영역의 이름으로 MP가 해당 공간 영역으로 일반화됨을 의미한다. ContainedMBR 연산을 통해 MBR들을 추출하면, ContainedArea 연산은 [그림 4]와 같이 각 MBR을 특정 크기의 그리드 셀(Grid Cell)로 분할하고, 실제 영역을 일부 또는 완전 포함하는 각 셀들을 X축이나 Y축을 기준으로 인접관계를 분석하여 그룹화한다(3). 이는 셀 그룹과 각 그룹 셀들에 대해서 MP의 포함 여부를 검사함으로써 MBR 영역에 대한 포함 관계 연산을 효율적으로 처리하기 위함이다. ContainedArea의 연산과정은 먼저, 각 영역 MBR에 대한 셀 그룹과 MP의 포함 여부를 비교하여 MP를 포함하는 셀 그룹을 탐색한다. MP를 포함하는 셀 그룹이 결정되면 해당 셀 그룹의 실영역 완전 포함 셀과 실영역 경계 포함 셀에 MP가 포함되는지 검사하여 완전 포함 셀이면 해당 영역 MBR의 이름을 반환하고, 영역 경계 포함 셀이면 ContainedGrid 연산을 호출하여 실영역 내에 MP가 존재하는지의 여부를 다시 검사한다. 만약 조건에 만족한다면 해당 영역 MBR의 이름을 반환한다. 이 때, 각 셀에 대한 MP의 포함 여부를 검사하기 위해서 [그림 5]와 같은 R\*-트리와 AGHT를 이용해 연산을 수행한다.



[그림 5] ContainedArea 연산을 위한 공간 탐색 트리

[그림 5]에서 레벨별 MBR 영역을 의미하는 Branch Node는 실제 영역을 일부 또는 완전 포함하는 셀들을 집 단화하여 생성한 셀 그룹 테이블을 갖는다. [그림 4]에서는 X축을 기준으로 하여 실영역을 포함하는 연속되는 셀들을 묶어 총 12개의 셀 그룹을 생성하게 된다. 또한 각 셀 그룹은 실영역에 완전 포함되는 셀과 실영역 경계 포함 셀의 정보를 저장한 각각의 테이블에 접근하기 위해서 셀 참조자(Pointer)를 갖는다. 이 때, 셀의 그룹화는 가능한한 MP와의 포함관계 연산 시 최소의 비교 횟수를 갖도록 그룹화한다.

Number of Cell Group in MBR<sub>k</sub> : K  
 Number of Cell in CellGroup<sub>j</sub> : R  
 Number of Cell in MBR<sub>i</sub> : Q  
 Set of Cell Group in MBR : MBR = {CellGroup<sub>1</sub>, CellGroup<sub>2</sub>, ..., CellGroup<sub>k</sub>}, 1 ≤ k ≤ K  
 Set of Cell in MBR<sub>i</sub> : MBR<sub>i</sub> = {Cell<sub>1</sub>, Cell<sub>2</sub>, ..., Cell<sub>q</sub>}, 1 ≤ q ≤ Q  
 Set of Cell in CellGroup<sub>j</sub> : CellGroup<sub>j</sub> = {Cell<sub>1</sub>, Cell<sub>2</sub>, ..., Cell<sub>r</sub>}, 1 ≤ r ≤ R, R ≤ Q  
 Relationship of Cell<sub>r</sub> and CompCell<sub>l</sub>, BoundCell<sub>p</sub> : If Cell<sub>r</sub> ≠ CompCell<sub>l</sub>, Then Cell<sub>r</sub> = BoundCell<sub>p</sub>, or  
 If Cell<sub>r</sub> ≠ BoundCell<sub>p</sub>, Then Cell<sub>r</sub> = CompCell<sub>l</sub>, ∀ r, l, p, r ≠ l ≠ p  
 Set of Cell in CellGroup<sub>k</sub> : CellGroup<sub>k</sub> = {[CompleteCell]<sub>a=0</sub>, [BoundaryCell]<sub>b=0</sub>}, 1 ≤ k ≤ R  
 If a = 0 Then b ≠ 0 or If b = 0 Then a ≠ 0  
 MBR of CellGroup : GupMBR  
 Bounding Points Set of GupMBR<sub>s</sub> :  
 GupMBR<sub>s</sub> = {GBP<sub>1,1,1,1</sub>, GBP<sub>1,1,1,2</sub>, GBP<sub>1,1,2,1</sub>, GBP<sub>1,1,2,2</sub>}, 1 ≤ s ≤ K  
 MBR of CompleteCell and boundCell : CellMBR  
 Bounding Points Set of CellMBR<sub>t</sub> :  
 CellMBR<sub>t</sub> = {CBP<sub>1,1,1,1</sub>, CBP<sub>1,1,1,2</sub>, CBP<sub>1,1,2,1</sub>, CBP<sub>1,1,2,2</sub>}, 1 ≤ t ≤ Q

input : ContMBR, MP, PrevA  
 Output : CurrA

```

Procedure ContainedArea(ContMBR, MP, PrevA)
Begin
    String CurrA;

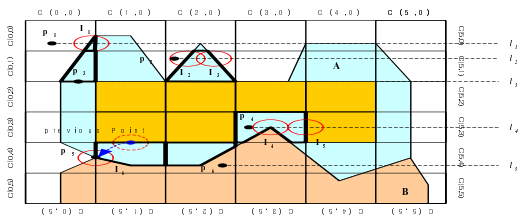
    For (i = 1 to ContMBR.Count)
        For (j = 1 to ContMBRi.CellGroup.Count)
            If ((MP contains ContMBRi.CellGroupj) || (MP touches ContMBRi.CellGroupj)) Then
                For (l = 1 to ContMBRi.CellGroupj.Cell.Count)
                    If ((MP contains ContMBRi.CellGroupj.Celll) || (MP touches ContMBRi.CellGroupj.Celll))
                        For (n = 1 to CellGroupj.CompCell.Count)
                            If (ContMBRi.CellGroupj.Celll == CellGroupj.CompCelln)
                                CurrA = ContMBRi.AreaNamen;
                            Else
                                For (m = 1 to CellGroupj.BoundCell.Count)
                                    If (ContMBRi.CellGroupj.Celll == CellGroupj.BoundCellm)
                                        CurrA = containedGrid(CellGroupj.BoundCellm, MP, PrevA);
                                    End If
                                End For
                            End If
                        End For
                    End If
                End For
            End If
        End For
    End For

    Return CurrA;
End
    
```

(그림 6) ContainedArea 연산

4.1.3 ContainedGrid 연산

ContainedGrid 연산은 ContainedArid 연산으로부터 MBR 에 대한 경계 포함 셀(boundCell)과 이동 객체의 위치 좌표점(MP), 이동 객체의 이전 위치 포함 영역(PrevA)을 전달받아 셀 경계를 포함하는 그리드의 셀 영역에 객체의 위치 좌표점이 포함되는지를 검사하는 공간 연산 함수이다. ContainedGrid 연산에 전달되는 셀의 정보 중 셀의 경계는 완전 포함 셀과 같이 셀의 경계 사각형(CellMBR)이 아닌 셀의 경계 사각형과 MBR 세그먼트 객체의 조합으로 구성된 새로운 셀 경계이다. 즉, [그림 7]에서 MP가 존재하는 경계 포함 셀 C(0,0), C(0,1), C(1,4), C(2,1), C(2,4), C(3,3)을 보면 두꺼운 실선으로 표현된 부분이 실제 셀 영역의 경계가 된다. 이러한 셀의 실 영역 내에 MP가 포함되는지에 대한 검사는 MP로부터 시작하는 단방향 직선 l을 이용하는 방법으로, 직선 l과 셀 영역 간선(경계선)들 간의 교차 여부를 통해 영역 내에 점이 포함되는지를 검사한다. 직선 l과 교차하는 영역 간선들의 수가 홀수 개이면 영역에 포함되고 0 이거나 짝수 개이면 영역 외부에 존재하는 것으로 판단한다. 단, 두 개 이상의 간선이 만나는 한 끝점을 직선 l이 통과할 경우, 각 간선의 또 다른 끝점이 l보다 높게 위치하면 교차하는 간선으로, 낮게 위치하면 교차하지 않는 간선으로 간주하며, 직선 l이 임의의 한 간선과 동일선 상에 위치할 경우에도 교차하지 않는다고 간주한다.



[그림 7] 영역 A의 셀과 직선 l과의 교차에 따른 포함 관계

[그림 7]에서 P<sub>2</sub>에서 시작하는 직선 l<sub>2</sub>는 셀 C(2,1)의 영역 간선들과 l<sub>2</sub>, l<sub>3</sub>에서 교차하게 되는데, l<sub>2</sub>, l<sub>3</sub> 모두 교차하는 간선으로 간주하여 l<sub>2</sub>는 교차하는 간선의 수가 각각 총 2개이므로 셀 C(2,1)의 영역 밖에 위치한다. 마지막 P<sub>3</sub>에서 시작하는 직선 l<sub>3</sub>의 경우 셀 C(0,1)의 한 영역 간선 상에 위치함으로 C(0,1)의 영역에 포함된다. 또한, P<sub>4</sub>에서 시작하는 직선 l<sub>4</sub>는 l<sub>4</sub>, l<sub>5</sub>에서 교차하게 되는데, l<sub>4</sub>의 경우 두 간선이 만나는 한 끝점을 직선 l<sub>4</sub>가 통과하지만 두 간선의 양 끝점 모두 직선 l<sub>4</sub>보다 높게 위치하지 않으므로 교차하지 않는 간선으로 간주하고, l<sub>5</sub>의 경우만을 교차하는 간선으로 간주한다. 마지막 P<sub>5</sub>는 두 개 이상의 영역들이 만나는 간선 상에 MP가 위치하는지를 검사하는 예로, 만약 영역 간선 상에 존재한다면 이동 객체가 이전에 위치했던 영역이 어디인지 판단하여 이전 영역에 포함된다고 간주한다. 즉, [그림 7]에서 P<sub>5</sub>는 영역 A와 영역 B 사이의 간선 상에 위치함으로 이전에 이동 객체가 위치한 영역인 A에 포함된다.

Boundary Line Set of BoundCell<sub>i</sub> : Edge = {Edge<sub>1</sub>, Edge<sub>2</sub>, Edge<sub>3</sub>, ..., Edge<sub>n</sub>}  
 Boundary Points Set of BoundCell<sub>i</sub> : BP = {BP<sub>1</sub>, BP<sub>2</sub>, BP<sub>3</sub>, ..., BP<sub>n</sub>}  
 End Points of Edge<sub>j</sub> and Edge<sub>n</sub> : Edge<sub>j</sub> = {BP<sub>j</sub>, BP<sub>j+1</sub>}, Edge<sub>n</sub> = {BP<sub>n</sub>, BP<sub>1</sub>}, 1 ≤ j < n  
 Horizontal pass across the MP : l

Input : BoundCell, MP, PrevA  
 Output : CurrA

Procedure ContainedGrid (BoundCell, MP, PrevA)

```

Begin
    String CurrA;
    Int count = 0;

    Edge = isCellEdge (BoundCell);

    For (i = 1 to Edge.Count)
        If (onEdge (Edgei, MP) == true) Then
            If (PrevA.Name ≠ Null) Then
                CurrA = PrevA.Name;
            Else
                CurrA = BoundCell.parentCellGroup.parentMBR.Name;
            End If
        Else
            If (Intersect (Edgei, l) == true)
                count = count + 1;
            End If
        End If
    End For

    If (count is odd) Then
        CurrA = BoundCell.parentCellGroup.parentMBR.Name;
    End If

    Return CurrA;
End
    
```

[그림 8] ContainedGrid 연산

4.2 이동 객체의 유효시간 일반화

하나의 시퀀스 MS={s<sub>1</sub>, s<sub>2</sub>, ..., s<sub>k</sub>}로 표현할 수 있다. 이때, s<sub>j</sub>=(at<sub>j</sub>, a<sub>j</sub>)이며, at<sub>j</sub>는 이동 객체가 a<sub>j</sub> 영역 내에 있을 때의 대표유효시간이고 a<sub>j</sub> ∈ A이다. 특정 영역(a<sub>i</sub>) 내 대표유효시간 AT<sub>i</sub>는 특정 영역에 들어갔을 때 처음 샘플링된 유효시간에서부터 나오기 전 샘플링된 유효시간까지의 평균유효시간으로 다음과 같이 정의된다.

$$AT_i = \frac{A_i Tout - A_i Tin}{2}$$

A<sub>i</sub>Tout : A<sub>i</sub> 영역에서 빠져나오기 전 샘플링된 유효시간  
 A<sub>i</sub>Tin : A<sub>i</sub> 영역에 들어갔을 때 처음 샘플링된 유효시간

[그림 ] 특정 영역내 시간

5. 결론 및 향후 연구과제

이동 객체의 위치 이력 데이터들은 공간상에서 x, y 좌표 값으로 표현되기 때문에 낮은 수준의 상세한 정보 형태를 띠고 있다. 하위 개념 수준으로 표현된 위치 데이터들로부터 직접적으로 지식화가 가능한 패턴들을 탐색하는 것은 매우 어려운 일이기 때문에 이동 객체의 연

속적 위치 변화를 보다 효과적으로 패턴화하기 위한 공간영역으로의 일반화 접근법이 필요하다. 따라서 본 논문에서는 이동 객체의 위치 속성에 대한 공간영역으로의 일반화를 위해 공간 개념 계층에 대한 정의와 R\*-Tree를 기반으로 레벨별 공간 영역을 그리드로 분할한 후 공간 영역의 탐색 트리를 생성하여 일반화를 수행하는 방법을 제시하였다. 이를 통해 일반화된 데이터 집합을 생성하여 이동 패턴 탐사를 보다 효과적으로 수행할 수 있도록 하였다. 공간 영역에 대한 정보는 MBR 간의 겹침의 최소화와 영역의 최소화에 기반한 최적화 인덱스인 R\*-Tree를 기반으로 구성하였으며, 효율적인 공간 영역으로의 일반화를 위해서 R\*-Tree의 비리프노드, 즉 공간 영역에 대한 MBR을 그리드로 분할하여 각 영역을 완전 포함 셀과 MBR 경계 포함 셀로 구분하여 Contains 공간 연산을 수행함으로써 MP[5,6]와 MPMine[13,14]에서 제시한 방법보다 효과적인 성능을 발휘하도록 하였다. 또한, Contains 연산은 공간 개념 계층에서 레벨 수준별 영역의 MBR에 대한 포함 여부를 검사하는 ContainedMBR 연산과 특정 셀의 영역 내에 포함되었는지를 검사하는 ContainedGrid 및 ContainedArea 연산을 제시하였다. 이러한 Contains 공간 연산을 통해 일반화된 각 공간 영역 데이터를 이용하여 이동 시퀀스를 생성하였으며, 이렇게 생성된 이동 시퀀스들은 의미있는 지식 추출을 위한 이동 패턴 탐사에 이용될 수 있고, 패턴 탐사를 통한 객체의 이동 추이 분석을 통해 다양한 형태의 서비스로 개발될 수 있다.

향후 연구과제로는 공간영역으로의 일반화 방법을 통해 생성된 일반화된 이동 객체 데이터와 이동 시퀀스를 이용하여 최적 이동 경로를 탐색하기 위한 마이닝 기법의 개발이 요구된다. 또한 최적 이동 경로 탐색을 위한 마이닝 기법을 이용하여 단위 시간동안 이동 객체가 순회해야 하는 지점들에 대한 스케줄링 경로 예측을 위한 마이닝 기법의 개발도 필요하다.

### 참 고 문 헌

[1] O. Sourina, and D. Liu, "Visual Interactive Clustering and Querying of Spatio- Temporal Data", Proc. of The International Conference on Computational science and its applications; ICCSA 2005, LECTURE NOTES IN COMPUTER SCIENCE(LNCS). Vol.3483, 2005.

[2] P. Kalnis, N. Mamoulis, S. Bakiras, "On Discovering Moving Clusters in Spatio-Temporal Data", Proc. of the 9th International Symposium on Spatial and Temporal Database(SSTD), LECTURE NOTES IN COMPUTER SCIENCE (LNCS). Vol.3633, 2005.

[3] N. Mamoulis, H. Cao, G. Kollios, M. Hadjieleftheriou, Y. Tao, and D. W. Cheung, "Mining, Indexing, and Querying Historical Spatio-Temporal Data", Proc. of the International Conference on Knowledge Discovery and Data Mining, 2004.

[4] P. Stolotz, and H. Nakamura, "Fast Spatio -Temporal Data Mining of Large Geophysical Datasets", Proc. of

the International Conference on Knowledge Discovery and Data Mining, 1995.

[5] J. D. Chung, O. H. Paek, J. W. Lee , K. H. Ryu, "Temporal Pattern Mining of Moving Objects for Location-Based Service", Proc. of the 13th International Conference on Database and Expert Systems Applications, pp.331-340, September 02-06, 2002.

[6] J. W. Lee, O. H. Paek, K. H. Ryu, "Temporal moving pattern mining for location-based service", The Journal of Systems and Software, Vol. 73. 2004.

[7] S. Ramaswamy, S. Mahajan and A. Silberschatz, "On the discovery of interesting patterns in association rules", the VLDB Conference, New York City, September 1998.

[8] H. Manniila, H. Toivonen, and A. I. Verkamo, "Discovery of frequent episodes in event sequences", Data Mining and Knowledge Discovery, 1(3), pp.259-289, November, 1997.

[9] R. Agrawal and R. Srikant, "Mining sequential patterns", In Proc. 11th International Conference on Data Engineering, 1995.

[10] R. Srikant and R. Agrawal, "Mining sequential patterns : Generalizations and Performance Improvements", International Conference on Extending Database Technology, Springer-verlag, 1996.

[11] E. Tsoukatos and D. Gunopoulos, "Efficient Mining of Spatio-Temporal Patterns", Proc. of the 7th International Symposium on Spatial and Temporal Database(SSTD), pp.425-442, 2001.

[12] J. Han, G. Dong, and Y. Yin, "Efficient Mining of Partial Periodic Patterns in Time Series Database", Proc. of The 11th International Conference on Data Engineering, 1999.

[13] 이준욱, 남광우, "이동객체 위치 일반화를 이용한 시공간 이동 패턴 탐사", 한국정보처리학회 논문지, 제 10-D권, pp.1103-1114, 제7호, 2003. 12.

[14] 이준욱, "지식 탐사 프레임워크 기반의 시공간 이동 패턴 탐사 기법", 박사학위논문, 충북대학교 대학원, 2003.

[15] 박지웅, "시공간 이동 패턴 추출을 위한 효율적인 알고리즘", 박사학위논문, 건국대학교 대학원, 2006.

[16] 이기영, "이동 객체 경로 탐색을 위한 시공간 클러스터링 기법", 박사학위논문, 건국대학교 대학원, 2005.

[17] J. Moreira, C. Ribeiro, and J. M. Saglio, "Representation and Manipulation of Moving Points : An Extended Data Model for Location Estimation", Cartography and Geographic Information System(CaGIS), ACSM, Vol.26, No.2, April 1999.