

# 이동 객체의 패턴 마이닝을 위한 위치 일반화 방법

고 현\*, 김광종\*, 이연식\*  
\*군산대학교 컴퓨터정보과학과  
e-mail:khyun001@kunsan.ac.kr

## Location Generalization Method for Pattern Mining of Moving Object

Hyun Ko\*, Kwang-Jong Kim\*, Yon-Sik Lee\*  
\*Dept of Computer Information Science, Kunsan National University

### 요 약

사용자들의 특성에 맞게 개인화되고 세분화된 위치 기반 서비스를 제공하기 위해서는 방대한 이동 객체의 위치 이력 데이터로부터 유용한 패턴을 추출하기 위한 시간 패턴 탐사가 필요하다. 기존의 시간 패턴 탐사 기법들은 이동 객체의 시간에 따른 공간 속성들의 변화를 충분히 고려하지 못하거나, 시공간 속성을 동시에 고려한 패턴 탐사는 가능하나 제약을 가진 공간 정보를 포함하는 패턴 탐사 문제에는 적용하기 어렵다. 따라서 이동 객체의 위치 이력 데이터들에 대한 시공간적 속성들을 동시에 고려하여 다양한 이동 패턴들 중 공간 제약을 만족하는 패턴들을 추출하기 위한 새로운 이동 패턴 탐사 기법이 요구된다. 이러한 패턴 탐사 기법의 개발을 위해서는 상세 수준의 위치 이력 데이터들을 공간 영역 정보 형태로 변환하는 위치 일반화 접근법이 필요하다. 이에 본 논문에서는 객체의 위치값과 공간 영역 간의 위상 관계를 고려하여 이동 객체의 위치 속성에 대한 공간영역으로의 일반화 방법을 제안한다. 이동 객체의 상세 수준의 위치 정보에서는 의미있는 패턴을 찾기가 어렵기 때문에 데이터 전처리 과정을 통해 일반화된 데이터 집합을 형성함으로써 효율적인 이동 객체의 시간 패턴 마이닝을 유도할 수 있다.

### 1. 서론

모바일 환경에서 모바일 기기 사용자들은 시간의 흐름에 따라 공간상을 끊임없이 이동하며 새로운 위치 정보를 지속적으로 생성하는 이동 객체(Moving Object)이다. 이동 객체의 위치 데이터를 이용해 효과적인 위치 기반 서비스를 제공하기 위해서는 방대한 위치 이력 데이터로부터 의미있는 지식인 유용한 패턴을 추출하기 위한 시간 패턴 탐사가 필요하다. 시간 패턴은 시간 속성을 가진 데이터로부터 찾아낼 수 있는 시간적인 지식으로 현재까지 시간 패턴의 유형별로 순차 패턴[1,2], 주기 패턴[3], 시간 관계[4]에 관한 많은 기법들이 제시되었다. 하지만, 기존의 시간 패턴 탐사 기법들은 시간 데이터의 대상으로 패턴 탐사를 수행하기 때문에 공간 속성들의 변화를 고려하지 않아 시간과 공간 속성을 동시에 고려해야 하는 이동 객체의 패턴 탐사 문제에 적합하지 않다. 또한 [5,6]와 같은 기법들은 시간과 공간 속성을 개별적으로 고려하는 문제가 있고, [7,8]은 시간과 공간 속성을 동시에 고려하여 패턴 탐사를 수행하지만 제약을 가진 공간 정보의 연속적인 변화에 대한 패턴 탐사에는 적용하기 어렵기 때문에 특정 공간상에서의 최적 경로 탐색이나 객체의 순회 지점들에 대한 스케줄링 경로 예측 문제 등에는 적합하지 않다.

이동 객체의 위치 이력 데이터들에 대한 시공간적 속성들을 고려하여 다양한 이동 패턴들 중 최적 이동 경로나 스케줄링 경로 예측을 위한 새로운 시간 패턴 탐사 기법의 개발을 위해서는 상세 수준의 위치 이력 데이터들을

의미있는 공간 영역 정보 형태로 변환하기 위한 방법이 필요하다. 이에 본 논문에서는 객체의 위치값과 공간 영역 간의 위상 관계를 고려하여 이동 객체의 위치 속성에 대한 공간영역으로의 일반화 방법을 제시한다. 이는 이동 객체의 위치 이력 데이터들을 일반화된 공간영역 정보 형태로 변환함으로써 지식화 가능한 패턴 탐사를 통해 의미있는 지식을 추출할 수 있도록 한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구로 시간 데이터 마이닝 기법들과 기존의 이동 객체 패턴 탐사 기법에 대해 기술하고, 3장에서는 위치 일반화를 위한 공간 연산 알고리즘을 제안한다. 4장에서는 제안한 기법을 구현하여 실험하고, 마지막 5장에서는 결론 및 향후 연구 과제를 제시한다.

### 2. 관련연구

시간 데이터 마이닝은 기존의 데이터 마이닝에 시간 개념을 추가하여 시간 의미와 시간 관계를 가지는 유용한 시간 지식을 탐사하기 위한 데이터 마이닝의 한 연구 분야이다. 현재까지 시간 데이터 마이닝에 대한 많은 연구가 수행되었으며, 기존의 연구들은 크게 시간 규칙(temporal rules) 탐사, 시퀀스 마이닝(sequence mining), 경향 분석(trend analysis) 등으로 분류된다. 시간 규칙 탐사는 연관 규칙 탐사나 분류 등의 기존 데이터 마이닝 기법에 시간 차원을 확장하여 주기 패턴, 캘린더 패턴 등과 같은 시간 패턴을 탐사하도록 확장한 기법으로 시간 연관 규칙, 시간

분류, 시간 특성화 등과 같은 기법들로 구분된다. 시퀀스 마이닝은 시퀀스 데이터로부터 의미있는 패턴을 탐사하는 기법으로 순차 패턴 탐사, 에피소드, 시간 확장 기법 등이 있다. 경향 분석은 시계열 데이터로부터 여러 형태의 경향을 탐사하는 기법으로 유사 시계열 탐사, 예외분석이 있다. 이러한 각 시간 데이터 마이닝 기법들은 단순히 시간 데이터를 대상으로 지식 탐사를 수행하기 때문에 시간의 흐름과 이동 객체의 위치 변화에 대한 관계성이 충분히 고려되지 않아 유용한 지식이 될 수 있는 시간에 따른 공간 변화의 패턴을 탐사하기 어렵다.

이외에도 시간 및 공간 속성을 가지는 데이터에 대해 지식 탐사를 하기 위한 연구도 일부 수행되었다. [5]는 공간 데이터에 대한 지식이 시간에 따라 어떻게 변화하는지를 탐사하는 것으로, 메타규칙(meta-rule)을 통해 규칙 속에서의 또 다른 규칙을 탐사하기 때문에 시간과 공간을 동시에 다루지 않는다. [6]은 특정 지역 내에서 일어난 환경적인 사건의 패턴을 탐사하는 것으로 각 지역은 일정한 크기를 가지는 것으로 한정하고 있어 실제계 공간 영역 상에서의 공간 객체 간의 관계 및 의미를 반영하기 어렵다는 문제점이 있다. 한편, [7,8]은 기존의 시간 패턴 마이닝 기법들이 이동 객체의 시간 및 공간 속성을 동시에 고려할 수 없는 문제를 해결하기 위한 방법으로, 이동 객체의 각 위치를 공간 영역으로 일반화하는 공간 연산을 적용함으로써 이동 객체의 시간 패턴을 탐사할 수 있도록 한 기법이다. 하지만 [7,8]의 경우 이동 객체의 공간 속성을 충분히 고려하여 공간 정보가 연속적으로 변화하는 이동 패턴 탐사는 가능하나, 제약이 있는 공간 영역에서의 위치 변화에 대한 의미있는 패턴 탐사를 수행하기는 매우 어렵다. 즉, 추출하고자 하는 패턴에 반드시 포함되어야 하는 공간 정보에 대한 제약이 없어 특정 지점들 사이의 최적 이동 경로 탐색 문제나 객체가 순회해야 지점들에 대한 스케줄링 경로 예측 문제 등에 적합하지 않다.

따라서 이동 객체의 위치 이력 데이터들에 대한 시공간 속성들을 동시에 고려하고 공간 제약에 대한 모든 조건들을 만족시킨 이동 패턴 탐사를 수행하기 위한 새로운 탐사 기법이 요구되며, 이러한 패턴 탐사를 위해 이동 객체의 연속적 위치 변화를 보다 효과적으로 패턴화하기 위한 공간영역으로의 일반화 방법이 필요하다. 일반적으로 이동 패턴 마이닝의 대상이 되는 객체의 위치 이력 데이터들은 샘플링된 시점에서 x, y 좌표값으로 표현되기 때문에 낮은 수준의 상세한 정보 형태를 띠고 있다. 이러한 하위 개념 수준의 위치 데이터들로부터 직접적으로 지식화 가능한 패턴들을 탐색하는 것은 매우 어렵기 때문에 하위 개념 수준으로부터 상위 개념 수준으로 올라가며 적절하게 데이터 집합을 추상화할 필요성이 있다.

**3. 위치 일반화를 위한 공간 연산 알고리즘**

**3.1 Contains 연산**

이동 객체의 위치 일반화는 객체의 위치 좌표점 MP가 공간 개념 계층에서 레벨 수준별 영역의 최소경계사각형(MBR : Minimum Bounding Rectangle)에 포함되었는지를 검사하는 단계와 특정 영역들의 경계에 위치하거나 또는 영역 내에 포함되었는지를 검사하여 위치 속성을 공간영역으로 일반화하는 단계를 거친다. 이 두 과정은 이동 객체가 스캔되는 위치 좌표점을 영역 범위로 변환하는 Contains 공간 연산을 통해 처리된다.

다음 (그림 1)은 이동 객체의 위치값을 특정한 공간 영역 범위로 일반화하는 Contains 공간 연산 알고리즘으로, 먼저, 시공간 질의 분석을 통해 결정된 공간 계층 레벨 수준(N)과 이동 객체의 위치 좌표점(MP), 객체의 이전 위

치 영역(PrevA)에 대한 정보들을 입력받아 최상위 레벨 수준의 전체 공간 영역을 입력된 하위 레벨 수준까지 최소 영역 사각형인 MBR로 분할한다. 이 때, 최상위 레벨 수준에서 하위 레벨 수준까지 MBR로 분할하는 이유는 시공간 질의 수준에 맞게 공간 개념 계층의 레벨 수준으로 이동 시퀀스를 생성하기 위함이다. Contains 연산은 질의 분석을 통해 N 레벨 수준까지 공간 영역을 반복해 분할하며 ContainedMBR 함수를 통해 MP가 N 레벨 수준별 영역 MBR에 포함되는지를 검사한다. 또한, MP가 포함된 모든 MBR들을 추출한 후 ContainedArea 함수를 이용하여 MP가 어느 영역의 경계 또는 내부에 위치하는지를 검사한다. 이 때, ContainedArea 함수의 결과로 반환되는 좌표점을 포함한 영역은 이동 시퀀스를 생성하기 위한 일반화된 영역이다.

```

Number of Spatial Concept Level : N
Set of Spatial Concepts Level : Level={L1,L2,...,Ln}
Number of Area MBR in Level L : M
Set of Area MBR in Level Li : Li={MBR1,MBR2,...,MBRm}, 1 ≤ i ≤ N
Bounding Point Set of Area MBR : MBRi={BP1,..., BP2,..., BP3,..., BP4,..., BP5,..., BP6,..., BP7,..., BP8,..., BP9,..., BP10}

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

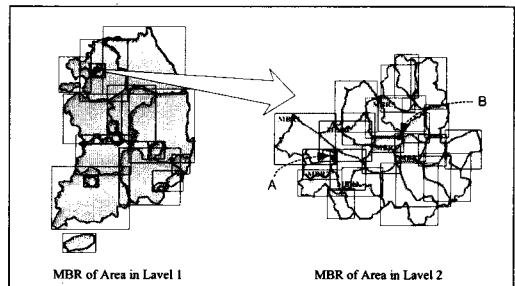
For (i=1 to N)
If (ContMBR ≠ ∅) Then
For (j=1 to ContMBR.Count) Then
Lj=ContMBR;
Low_CA=ContainedMBR(MP,Lj);
End For
ContMBR=Low_ContMBR;
Else
ContMBR=ContainedMBR(MP,Li);
End If
End For

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

(그림 1) Contains 공간 연산 알고리즘

**3.2 ContainedMBR 연산**

위치 일반화 위한 두 단계 중 먼저, 첫 번째 단계는 공간 개념 계층의 레벨 수준에 따라 전체의 공간 영역을 레벨 수준별 영역의 MBR로 분할하여 각 MBR에 이동 객체의 위치 좌표점이 포함되었는지를 검사하는 단계이다. 다음 (그림 2)와 같이 이동 객체가 A에서 B로 이동 시 이동 경로를 탐색하려할 경우, 이동 시퀀스의 생성에 있어 레벨 0 수준에서는 시퀀스의 생성이 불가능하므로 레벨 1 수준에서 시퀀스를 생성해야 한다. 따라서 레벨 0 수준의 영역을 레벨 1 수준의 하위 영역으로 분할한 후 각 영역에 대한 MBR을 생성하고 A와 B가 포함되는 각각의 MBR을 추출한다. A는 MBR1, MBR2, MBR3, MBR4에 포함되고 B는 MBR5, MBR6, MBR7에 포함된다.



(그림 2) 레벨 수준별 공간 영역에 대한 MBR

다음 (그림 3)의 ContainedMBR 연산은 이동 객체의 위치 좌표점(MP)과 각 레벨 수준별 영역 MBR 집합(L<sub>i</sub>)을 입력받아 MP가 각 MBR에 포함되는지를 검사하여 해당 MBR들의 집합(ContMBR)을 반환하는 공간 연산이다.

```

Vector ContainedMBR (MP, Li)
Begin
  For (j=1 to Li.Count)
    If (MP properly_contains Li.MBRj) || (MP meets Li.MBRj) Then
      ContMBR[ContMBR.Count+1]=Li.MBRj;
    End If
  End For

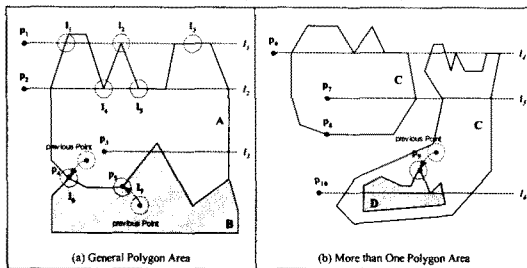
  Return ContMBR;
End
    
```

(그림 3) ContainedMBR 연산

3.3 ContainedArea 연산

이동 객체의 위치 일반화에서 두 번째 단계는 첫 번째 단계를 거쳐 추출한 레벨 수준별 영역 MBR들의 실영역에 대해서 MP가 영역의 경계에 위치하는지 또는 영역 내에 포함되는지를 검사하여 객체의 위치값을 영역 범위로 일반화하는 단계이다. 이 방법은 먼저, 두 개 이상의 영역들이 만나는 간선(경계선) 상에 MP가 위치하는지를 검사한다. 만약 영역 간선 상에 MP가 존재한다면 이동 객체가 이전에 위치했던 영역이 어디인지를 추적하여 이전 영역에 포함된다고 간주한다. (그림 4-(a))에서 P<sub>4</sub>와 P<sub>5</sub>의 경우, 이전의 위치 영역인 A와 B에 포함된다.

한편, 이동 객체의 위치 좌표점이 영역 내부에 포함되는지에 대한 검사는 객체의 위치 좌표점으로부터 시작하는 단방향 직선 l을 이용하는 방법으로, 직선 l과 영역 간선들 간의 교차 여부를 통해 영역 내에 점이 포함되는지를 검사한다. 직선 l과 교차하는 영역 간선들의 수가 홀수 개이면 영역에 포함되고 짝수 개이면 영역 외부에 존재하는 것으로 판단한다. 단, 두 개 이상의 간선이 만나는 한 끝점을 직선 l이 통과할 경우(I<sub>2</sub>I<sub>4</sub>I<sub>5</sub>), 각 간선의 또 다른 끝점이 l보다 높게 위치하면 교차 간선(I<sub>4</sub>I<sub>5</sub>)으로, 낮게 위치하면 비교자 간선(I<sub>2</sub>)으로 간주하며, 직선 l이 임의의 한 간선과 동일선 상에 위치할 경우(I<sub>3</sub>)에도 비교자 간선으로 간주한다. (그림 4-(a))에서 직선 l은 I<sub>1</sub>의 경우 교차 간선으로, I<sub>2</sub>와 I<sub>5</sub>는 비교자 간선으로 간주하여 총교차 간선의 수가 2개이므로 P<sub>1</sub>은 A 영역 밖에 위치한다. 또한, 직선 l<sub>2</sub>에서 I<sub>4</sub>, I<sub>5</sub>의 경우, 간선의 두 점들 중 교차점이 아닌 또 다른 끝점이 직선 l<sub>2</sub>보다 높은 곳에 위치하여 교차 간선으로 인정된다. 따라서 l<sub>2</sub>의 총교차 간선 수는 6개로 P<sub>2</sub>는 A 영역 밖에 존재하게 된다. 마지막 직선 l<sub>3</sub>의 경우, 간선과의 교차 수가 3개이므로 P<sub>3</sub>은 영역 A에 포함된다.



(그림 4) 직선 l 과의 교차에 따른 점과 영역의 포함 관계

(그림 4-(b))는 여러 개의 다각형으로 구성된 영역에서 직선 l과 각 영역 간선들과의 포함 관계를 나타낸 것으로, 위와 동일한 방법을 사용하여 영역 포함 여부를 판별한다.

따라서 l<sub>4</sub>와 l<sub>5</sub>의 경우 교차 간선의 수가 각각 6개와 3개로, P<sub>6</sub>는 영역의 외부에 P<sub>7</sub>는 영역 내부에 위치한다. 또한 P<sub>8</sub>의 경우 임의의 영역 간선 상에 존재함으로 해당 영역에 포함된다. 이 외에도 특정 영역 내에 또 다른 영역을 완전 포함하는 형태의 영역 객체에 대한 점 객체의 포함 여부 검사는 내부 영역 간선들과의 교차는 모두 무시하고 외부 간선들과의 교차 수에 따라 포함 여부를 판별한다. 즉, 직선 l<sub>6</sub>은 영역 C와 D를 모두 통과하나 D 영역 간선들과의 교차는 모두 무시되고 단지 C 영역 외부 간선들과의 교차만이 인정되어 총 2개의 교차 간선이 존재하므로 P<sub>10</sub>은 C 외부에 존재한다. P<sub>9</sub>는 이전 점의 위치가 C 영역 내에 존재했었으므로 P<sub>9</sub>는 영역 C에 포함된다.

```

Area contained MBR : A=(E1,E2,E3...En), A=(AP1,AP2,AP3...APn)
End Points of Area Edge i : Ei=(APi,APi+1), En=(APn, AP1), 1 ≤ i < n
Horizontal Pass across the MP : l
    
```

```

String ContainedArea (ContMBR, MP, PrevA)
Begin
  String A_name;
  Int count=0;

  For (i=1 to ContMBR.Count)
    A=IsAreaEdge(ContMBRi);
    For (j=1 to A.Count)
      If (onEdge(A.Ej,MP) == true) Then
        A_name=PrevA.Name;
        Break;
      Else
        If (Intersect(A.Ej,l) == true) Then
          count=count+1;
        End If
      End If
    End For

    If (count is odd) Then
      A_name=A.Name;
    End If
  End For

  Return A_name;
End
    
```

(그림 5) ContainedArea 연산

(그림 5)의 ContainedArea 연산은 MP를 포함하는 MBR 집합(ContMBR), 객체의 위치 좌표점(MP), 객체의 이전 위치 영역(PrevA)을 전달받아 각 영역 MBR에 대응하는 실영역들이 MP를 포함하는지 검사하는 공간 연산이다. ContainedArea 연산은 isAreaEdge 함수를 통해 특정 레벨 수준의 영역 MBR로 구성된 MBR 집합으로부터 각 MBR에 대한 실제 영역의 간선 정보들을 추출한다. 특정 영역에 대한 간선 정보가 추출되면, onEdge 함수는 각각의 간선 상에 MP가 위치하는지를 검사한다. 만약 MP가 간선 상에 존재한다면 이전에 MP가 위치했던 영역의 이름을 반환하고, 그렇지 않다면 Intersect 함수에 의해 교차 간선의 수를 계산한다. Intersect 함수는 MP에서부터 시작하는 직선 l이 영역 간선들과 교차하는지에 대한 여부를 판별하는 공간 연산 함수이다. 또한, ContainedArea 연산은 Intersect 함수에 의해 판별된 교차 간선의 총 개수가 짝수인지 혹은 홀수인지를 계산하여 홀수일 경우 MP가 해당 영역에 포함된다고 간주하여 영역 이름을 반환한다.

4. 알고리즘 구현 및 실험

위치 일반화 방법은 데이터베이스 정렬 단계를 거친 후 제한된 공간 연산 알고리즘을 통해 위치 일반화를 수행하여 이동 시퀀스를 생성하게 된다. 이동 객체 데이터베이스는 데이터 마이닝을 위해 객체 식별자를 주기로, 트랜잭션 시간을 보조키로 정렬한다. 다음 <표 1-(a)>는 객체 식별

자와 트랜잭션 시간에 의해 정렬된 데이터베이스이고, <표 1-(b)>는 객체의 공간값 속성이 어떠한 영역에 포함되는지 Contains 공간 연산을 적용하여 일반화된 영역으로 변환한 예이다. <표 1-(c)>는 일반화된 영역을 시간 간격을 고려하여 이동 시퀀스로 생성한 것이다.

<표 1> Contains 연산을 통한 일반화 영역으로의 변환  
(a) 정렬된 데이터베이스 (b) 위치 일반화 예

OID	VT	X	Y	OID	VT	Area
1	2005/11/20/13/10	21	73	1	2005/11/20/13/10	S
	2005/11/20/13/22	76	75		2005/11/20/13/22	A
	2005/11/20/13/35	113	98		2005/11/20/13/35	B
	2005/11/20/13/53	187	101		2005/11/20/13/53	C
	2005/11/20/14/20	234	80		2005/11/20/14/20	F
2	2005/11/21/12/45	80	78	2	2005/11/21/12/45	A
	2005/11/21/12/55	115	94		2005/11/21/12/55	B
	2005/11/21/13/13	182	92		2005/11/21/13/13	C
	2005/11/21/13/29	179	50		2005/11/21/13/29	E
	2005/11/26/09/30	179	50		2005/11/26/09/30	E
2005/11/26/09/51	236	70	2005/11/26/09/51	F		
3	2005/11/22/17/15	23	77	3	2005/11/22/17/15	S
	2005/11/22/17/28	72	77		2005/11/22/17/28	A
	2005/11/22/17/44	116	48		2005/11/22/17/44	D
	2005/11/22/17/50	182	43		2005/11/22/17/50	E
	2005/11/22/18/19	241	67		2005/11/22/18/19	F
4	2005/11/22/18/03	17	74	4	2005/11/22/18/03	S
	2005/11/22/18/18	73	78		2005/11/22/18/18	A
	2005/11/22/18/37	118	47		2005/11/22/18/37	D
	2005/11/22/18/48	124	99		2005/11/22/18/48	B
	2005/11/24/09/50	124	99		2005/11/24/09/50	B
2005/11/24/10/12	174	49	2005/11/24/10/12	E		
2005/11/24/10/28	181	102	2005/11/24/10/28	C		

(c)이동 시퀀스(트랜잭션 데이터)

OID	Moving	OID	Moving
1	<S A B C F>	3	<S A D E F>
2	<A B C E>, <E F>	4	<S A D B>, <B E C>

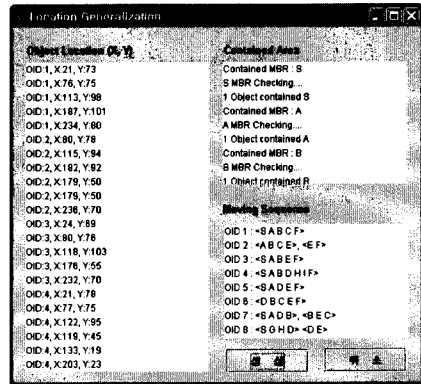
<표 1-(b)>와 같이 이동 객체의 위치 속성이 Contains 연산을 통해 공간영역으로 일반화 되면 이동 객체의 이동 변화를 시간에 따라 순차적으로 나열하여 이동 시퀀스를 생성할 수 있으나, 이러한 영역의 순차리스트는 사건(영역 변화) 사이의 시간 간격(interval)이 고려되지 않아 패턴 마이닝을 위한 트랜잭션인 이동 시퀀스로 사용하기에는 문제가 있다. 따라서 시퀀스를 구성하는 영역 간에 최대 시간 간격에 대한 제약조건을 두어 이를 만족해야만 의미 있는 이동 시퀀스로 생성될 수 있도록 해야 한다. 즉, 이동 객체의 공간 속성에 대한 샘플링 시간을 검사하여 특정 영역에 머문 시간이 최대 시간 간격 max\_gap을 초과하면 초과 이전까지의 영역 순차리스트에 대한 이동 시퀀스와 초과 이후의 이동 시퀀스로 분리하여 생성한다.

가령, <표 1-(b)>에서 이동 객체 2의 이동 경로는 영역 간을 이동한 시간 간격을 고려하지 않았을 경우 A→B→C→E→F 로 표현될 수 있다. 만약 영역 간의 이동 시간 간격(max\_gap)이 최대 1일이라면 영역 E에서 멈추었다가 다시 F를 향해 출발하는데 걸린 시간이 5일이기 때문에 이동 경로 A→B→C→E→F는 <A B C E>와 <E F>, 두 개의 시퀀스로 분리되어야 한다. 마찬가지로 이동 객체 4의 이동 시퀀스는 <S A D B>와 <B E C>로 분리된다.

다음 (그림 6)은 Contains 공간 연산 알고리즘을 구현을 구현하여 <표 1-(a)>의 이동 객체 위치 정보에 대한 일반화된 영역 및 이동 시퀀스를 생성한 결과이다.

5. 결론 및 향후 연구과제

이동 객체의 위치 이력 데이터들은 공간 상에서 x,y 좌표값으로 표현되기 때문에 하위 수준의 상세한 정보 형태를 띠고 있다. 하위 개념 수준으로 표현된 위치 데이터들로부터 직접 지식화 가능한 패턴들을 탐사하는 것은 매우 어려운 일이므로 이동 객체의 연속적 위치 변화를 보다



(그림 6) 위치 일반화 실행 결과

효과적으로 패턴화하기 위한 공간영역으로의 일반화 접근법이 필요하였다. 따라서 본 논문에서는 Contains 공간 연산을 이용해 이동 객체의 위치 속성에 대한 공간영역으로의 일반화 방법을 제시하였고, 일반화된 데이터 집합으로부터 이동 시퀀스를 생성하여 패턴 탐사를 보다 효과적으로 수행할 수 있도록 하였다. Contains 연산으로는 공간 개념 계층에서 레벨 수준별 영역의 MBR에 대한 포함 여부를 검사하는 ContainedMBR 연산과 MBR에 대응한 실 영역 내에 포함되었는지를 검사하는 ContainedArea 연산을 제시하였다. 또한, 일반화된 각 공간 영역 데이터를 시간 순차에 따라 리스트로 형성한 후 시간 간격 제약을 적용하여 이동 시퀀스를 생성하였다. 이렇게 생성된 이동 시퀀스들은 의미있는 지식 추출을 위한 이동 패턴 탐사에 이용될 수 있으며, 패턴 탐사를 통한 객체의 이동 추이 분석을 통해 다양한 형태의 서비스도 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.

참고문헌

- [1] R. Agrawal and R. Srikant, "Mining sequential patterns", In Proc. 11<sup>th</sup> International Conference on Data Engineering, 1995.
- [2] R. Srikant and R. Agrawal, "Mining sequential patterns : Generalizations and Performance Improvements", International Conference on Extending Database Technology, Springer-verlag, 1996.
- [3] B. Ozden, S. Ramaswamy, and A. Silberschatz, "Cyclic association rules", Int'l Conference on Data Engineering, Orlando, 1998.
- [4] J. Han, G. Dong, and Y. Yin, "Efficient Mining of Partial Periodic Patterns in Time Series Database", Proceedings of the 11th International Conference on Data Engineering, 1999.
- [5] T. Abraham and J. F. Roddick, "Discovering Meta-rules in Mining Temporal and Spatio-temporal data", Proc. of the International Database Workshop, Data Mining, Data Warehousing and Client/Server Databases, (IDW'97), pp.30-41, 1997.
- [6] 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.
- [7] 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, p.331-340, September 02-06, 2002.
- [8] 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.