

시퀀스 빈발도와 가중치를 이용한 최적 이동 패턴 탐사[☆]

Optimal Moving Pattern Mining using Frequency of Sequence and Weights

이 연 식*
Yonsik Lee

박 성 숙**
Sungsook Park

요 약

사용자들의 특성에 맞게 개인화되고 세분화된 위치 기반 서비스를 개발하기 위한 목적으로 시공간 상에서 발생하는 이동 객체의 다양한 패턴들 중 의미있는 유용한 패턴을 추출하기 위한 시공간 패턴 탐사가 필요하다. 이에 본 논문에서는 방대한 이동 객체의 이력 데이터로부터 패턴 탐사를 통해 실세계에 적용 가능한 위치 기반 서비스의 개발에 대한 응용으로, STOMP(F)(25)에서 정의한 최적의 이동 패턴을 탐사하는 문제들을 기반으로 시간 및 공간 제약을 갖는 패턴을 추출하기 위한 새로운 탐사 기법인 STOMP(FW)를 제안한다.

제안된 기법은 패턴 빈발도만을 이용한 기존 연구(STOMP(F)(25))에 가중치(거리, 시간, 비용 등)를 복합적으로 이용하는 패턴 탐사 방법으로, 특정한 지점들 사이를 이동한 객체의 이동 패턴들 중 패턴 빈발도가 특정 임계치 이상이고 가중치가 가장 적게 소요되는 이동 패턴을 최적 경로로 결정하는 방법이다. 제안된 방법의 패턴 탐사는 경험적인 이동 이력을 사용함으로써 기존의 최적 경로 탐색 기법들(A*, Dijkstra 알고리즘)이나 빈발도만을 이용한 방법들 보다 접근하는 노드 수가 상대적으로 적어 보다 빠르고 정확하게 최적 패턴을 탐색할 수 있음을 보인다.

Abstract

For developing the location based service which is individualized and specialized according to the characteristic of the users, the spatio-temporal pattern mining for extracting the meaningful and useful patterns among the various patterns of the mobile object on the spatio-temporal area is needed. Thus, in this paper, as the practical application toward the development of the location based service in which it is able to apply to the real life through the pattern mining from the huge historical data of mobile object, we are proposed STOMP(using Frequency of sequence and Weight) that is the new mining method for extracting the patterns with spatial and temporal constraint based on the problems of mining the optimal moving pattern which are defined in STOMP(F)(25).

Proposed method is the pattern mining method compositively using weighted value(weights) (a distance, the time, a cost, and etc) for our previous research(STOMP(F)(25)) that it uses only the pattern frequent occurrence. As to, it is the method determining the moving pattern in which the pattern frequent occurrence is above special threshold and the weight is most a little bit required among moving patterns of the object as the optimal path. And also, it can search the optimal path more accurate and faster than existing methods(A*, Dijkstra algorithm) or with only using pattern frequent occurrence due to less accesses to nodes by using the heuristic moving history.

☞ Keyword : 시공간 패턴 탐사(Spatio-Temporal Pattern Mining), 최적 경로 탐색(Optimal Path Search), 시퀀스 빈발도(Frequency of Sequence), 가중치(Weight), STOMP(FW) 알고리즘(Spatial-Temporal Optimal Moving Pattern(FW) Algorithm)

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

** 정 회 원 : 한국폴리텍 V 김제 대학 교수
joara@kopo.ac.kr

[2008/12/08 투고 - 2008/12/23 심사 - 2009/04/14 심사완료]

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

1. 서 론

이동 객체의 시간에 따른 위치 변화 이력(history)을 이용한 위치 기반 서비스를 제공하기 위해서는 이력 데이터들로부터 특정한 정보나 특성을 추출하여 서비스화 할 수 있는 방법이 요구된다. 현재까지는 객체의 시간에 따른 위치 변화

성을 고려하여 객체의 위치 및 궤적을 추적하는 방법들[18,19,20]과 객체의 현재 위치, 속도, 방향 정보를 이용하여 미래의 위치를 예측하기 위한 방법들[1,2,3,4]이 제안되었다. 하지만 이러한 방법들은 이력 데이터를 탐색하여 예측 가능한 결과를 생성하는 방법만을 제시한 것일 뿐 이력 데이터 자체 혹은 데이터들 간의 관계에 내포된 의미 있는 지식 추출을 위한 패턴 탐사에는 한계가 있다. 따라서 이동 객체의 동적인 위치 변화에 따른 특성 분석을 통해 유용한 정보를 추출하기 위한 패턴 탐사 방법이 필요하다. 시공간 이동 패턴 탐사는 연속된 시간상에서 발생하는 객체의 다양한 이동 패턴들 중 위치변화에 따른 독특한 경향이나 반복적이고 공통적인 의미있는 패턴을 탐사하기 위한 기술이다. 현재까지 연구된 대부분의 탐사 기법들은 순차패턴[7,8,9,10,21,22,23,24]이나 주기패턴[5,6]과 같이 반복적이고 주기적으로 발생하는 빈발 이동 패턴을 탐사하기 위한 방법들로, 다양한 응용 분야에 적용할 수 있는 위치 기반 서비스의 개발을 목적으로 하기 보다는 단순히 효율적인 패턴 탐사를 수행하기 위한 성능 향상에 주로 목적을 두고 있다. 또한 기존의 탐사 기법들은 시간 및 공간 한정자(qualifier)에 의해 제한된 범위의 시간 및 공간 영역에서 발생하는 모든 이동 패턴들 중 불특정한 빈발 이동 패턴을 탐사하기 때문에 시간 및 공간 한정자 이외의 또 다른 시간 및 공간 제약 조건을 가진 시공간 이동 패턴을 탐사하는 문제에 대해서 능동적으로 대처하기 어렵다[25]. 따라서 이러한 문제를 해결하여 보다 효율적으로 패턴 탐사를 수행하는 새로운 탐사 기법이 필요하다.

이에 본 논문에서는 방대한 시공간 이동 데이터 집합으로부터 다양한 이동 패턴들 중 복잡한 시간 및 공간 제약 조건을 만족하고, 실세계에서의 최적 경로에 해당하는 최적 이동 패턴을 탐사하기 위하여 패턴 빈발도만을 이용하여 최적 경로를 탐색하는 기존의 연구[25]를 기반으로 거리, 시간, 비용 등의 가중치를 고려한 최적경로를

탐색하는 STOMP(FW) (Spatial-Temporal Optimal Moving Pattern (Frequency&Weight)) 알고리즘을 제시한다. 제안된 알고리즘은 특정한 지점들 사이를 이동한 객체의 패턴들 중 패턴 빈발도가 특정 임계치 이상의 빈도를 보이고, 가중치(거리, 시간, 비용, etc)가 가장 적게 소요되는 이동 패턴을 최적 경로로 결정하는 방법으로, 경험적인 이동 이력을 사용함으로써 기존의 최적 경로 탐색 기법들보다 접근하는 노드 수가 상대적으로 적어 보다 빠르고 정확하게 최적 패턴을 탐색할 수 있다.

본 논문의 구성은 2장에서 관련연구로 기존의 이동 패턴 탐사 기법들에 대해 기술한다. 3장에서는 빈발도 및 가중치 기반의 최적 이동 패턴 탐사 문제에 대한 정의와 최적 이동 패턴 탐사 과정을 단계적으로 제시하고, STOMP(FW) 알고리즘을 설계 및 구현한다. 4장에서는 실험을 통해 제안된 탐사 기법의 성능을 평가하고, 마지막 5장에서는 결론 및 향후 연구 과제를 제시한다.

2. 관련연구

시공간 이동 패턴 탐사는 연속적인 시간의 흐름에 따라 이동하는 시공간 객체의 패턴들 중 객체의 위치 변화에 따른 공통적인 경향이나 의미 있는 패턴을 탐사하는 기법이다. 현재까지 연구된 시공간 이동 패턴 탐사 기법들로는 이동 객체가 이동하는 위치 변화의 패턴을 탐사하는 이동 패턴 탐사[5,6,7,8,9,10,21,22,23,24]와 이동 객체의 위치 변화에 있어 집단화되는 패턴을 탐사하는 그룹 패턴 탐사[11,12], 이동 객체가 군집을 형성하며 이동하는 움직임에서 패턴을 탐사하는 모션 패턴 탐사[13,14] 기법 등이 있다.

Mamoulis[6]은 이동 객체의 이력 데이터 집합으로부터 다양한 이동 객체의 이동 패턴들 중 주기적으로 발생하는 빈발한 이동 패턴을 추출하기 위한 방법으로 STPMine1 (Spatial-Temporal Pattern Mine1)과 STPMine2 알고리즘을 제시하였다. Apriori 계열의 알고리즘을 변형한 STPMine1 알고

리즘은 빈발 이동 패턴 탐사 시 후보 패턴의 길이가 길어질수록 빈번한 데이터베이스 스캔으로 인해 탐사 수행 성능이 저하되는 문제를 가지고 있다. 이러한 문제로 Mamoulis[6]은 STPMine2에서 데이터베이스의 접근을 최소화하기 위해서 메모리상에 Max_Subpattern 트리를 생성하여 효율적인 패턴 탐사가 이루어지도록 하였다. 또한, STPMine2와 유사한 방법으로 Kim[5]도 이동 패턴 트리를 생성하여 대용량의 시공간 데이터 집합으로부터 이동 객체의 이동 패턴을 효율적으로 추출하기 위한 STMPE(Spatio-Temporal Moving Pattern Extracting) 알고리즘을 제시하였다. STMPE는 이동 객체의 이력 정보를 분석하여 시간 및 공간 정보를 모두 포함하는 일반화된 이동 패턴을 추출하는 알고리즘으로, 시공간 데이터 일반화 기능과 이동 패턴 트리 생성 기능을 지원함으로써 패턴 탐사 시 데이터를 표현하는데 필요한 저장공간의 크기를 감소시키고 데이터베이스의 스캔 횟수를 감소시켜 탐사 수행 시간을 최소화하도록 하였다. 이러한 연구들은 패턴 탐사를 통해 추출되는 지식들이 실제세계의 다양한 분야에서 제공되도록 위치 기반 서비스 개발을 목적으로 수행되었기 보다는 패턴 탐사 수행 성능의 향상을 목적으로 연구되었기 때문에 특정 응용 분야에 적용가능한 새로운 위치 기반 서비스에 대한 개발이 필요하다.

하지만 일부에서는 이동 객체의 빈발한 이동 패턴을 추출하여 특정 분야에 적합한 위치 기반 서비스로 활용하기 위한 연구들이 시도되었다. Yavas[10]는 PCS(Personal Communication System) 네트워크 안에서 사용자의 이동성 예측(mobility prediction)을 위해서 이동성 패턴을 추출하는 UMP(User Mobility Pattern) 마이닝 기법을 제안하였다. 또한 한선영[24]은 다양한 이동 객체들의 공통적인 이동 패턴을 분석하여 그 패턴에 따라 미술관이나 박물관에서의 전시물 배치, 쇼핑몰에서의 새로운 제품 판촉이나 상품 배치 등과 같은 위치 기반 서비스를 제공하기 위한 패턴 탐사 기

법인 Apriori-MSP 알고리즘을 제시하였다. 이러한 연구들처럼 시공간 객체의 주기적이거나 반복적인 빈발 이동 패턴의 추출을 통해 실제세계의 보다 다양한 분야에 적용할 수 있는 새로운 위치 기반 서비스의 개발이 아직은 부족한 실정임으로 이에 대한 연구가 지속적으로 필요하다.

한편 지금까지 고찰한 기존의 이동 탐사 기법들은 시간 및 공간 한정자(qualifier)에 의해 제한된 시간 및 공간 영역에서 발생한 모든 이동 패턴들 중 불특정한 빈발 이동 패턴들만을 탐사하기 때문에 시간 및 공간 한정자 이외의 또 다른 시간 및 공간 제약 조건을 가지는 의미있는 이동 패턴을 탐사하기에는 문제가 있다.

3. 빈발도 및 가중치 기반의 최적 이동 패턴 탐사 기법

3.1 최적 이동 패턴 탐사 문제 정의

최적 이동 패턴 탐사는 방대한 이동 객체의 이력 데이터들로부터 특정한 지점들 간을 이동하는 다양한 패턴들 중 가장 최적의 비용을 소요하는 이동 패턴을 추출한다. 이러한 최적 이동 패턴을 탐사하기 위해 패턴 빈발도와 가중치를 이용할 수 있다. 빈발도와 가중치를 이용한 최적 이동 패턴 탐사 방법은 객체의 다양한 이동 패턴들에 대해 빈발도(지지도)를 추출한 후 지지도 임계치를 적용하여 최소지지도 이상이고 가중치(거리, 시간, 비용 등)가 가장 적게 소요되는 이동 패턴을 탐사하는 방법이다. 제안된 방법의 기본 개념은 특정 지점 간을 이동한 객체의 패턴들 중 빈번하게 발생할 패턴이 대체로 최적의 비용을 소요하는 패턴일 확률이 높다는 가정을 기반으로 하지만 가장 빈번한 패턴이 반드시 최적의 비용을 소요한다고는 단정할 수 없다. 또한, 지지도에 대한 고려뿐만 아니라 사용자가 의도하는 최적 패턴에 대한 평가 요소(거리, 시간, 비용 등)가 최소일 때 최적 경로로 선택될 확률이 높다. 따라서 빈발도

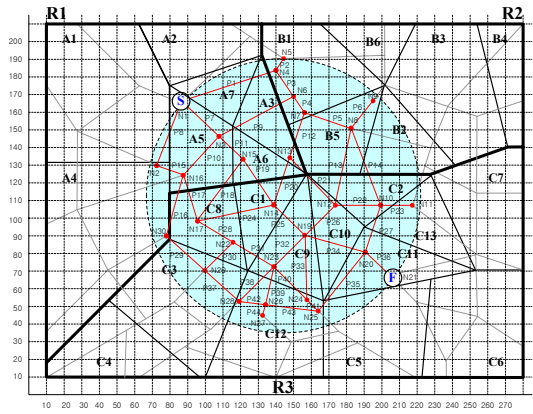
와 가중치를 이용한 최적 이동 패턴 탐사 문제를 정의하기에 앞서 최적 이동 패턴에 대해 다음과 같이 가정한다.

- 가정 1. 임의의 시간대에 특정지점 간을 이동하는 객체의 패턴들 중 가장 빈번하게 발생하는 패턴이 반드시 최적의 비용을 소요하는 것은 아니다.
- 가정 2. 특정지점들 사이의 이동 패턴들 중 지지도(빈발도)가 낮은 패턴이 최적 경로일 확률은 매우 낮다.
- 가정 3. 이동 객체의 이동에 있어 시간이나 이동 비용 등의 가중치가 최소일 때 최적 경로일 확률이 높다.

실세계에서 객체들은 특정지점에서 목적지까지 가장 빠르고 가장 적은 비용을 소요하는 경로를 통해 이동하려는 성향을 가진다. 빈발도 기반의 최적 패턴 탐사 방법[25]과 같이 이동 객체의 패턴들 중 가장 빈번하게 발생하는 패턴을 최적 경로로 결정하는 방법은 특정 시간대에 객체의 과중 밀집 또는 개인별 성향 차이 등의 문제로 인해 최적의 비용을 소요하는 경로가 될 수 없다. 하지만 객체의 빈번한 이동 경로가 아닌 경로 또한 특정한 외부적인 제한(이동경로 용적량, 제한 속도, etc)으로 인해 최적 경로로 결정되기는 극히 희박하다. 따라서 [가정 1]과 [가정 2]를 만족하는 패턴들 중 [가정 3]의 항목을 동시에 만족하는 패턴이 가장 최적 경로일 확률이 높다. 이러한 조건에 따라 빈발도와 가중치를 기반으로 한 최적 이동 패턴 탐사 문제를 정의하면 다음과 같다.

- 정의 1. 이동 객체 데이터베이스 MD, 사용자가 지정한 특정지점들 S와 F(공간제약), 시간기간 T1과 T2(시간제약), 최소지지도 min_sup 이 주어졌을 때, 공간 및 시간제약을 만족하는 이동 시퀀스 집합으로부터 최소지지도를 만족하는 S에서 F까지의 모든 빈발한 부분 시퀀스들을 추출하여 가중치가 가장 적은 시퀀스들의 집합을 탐색하는 것이다. 단, 부분 시퀀스는 S에서 F까지를 도달하는 하나의 시퀀스에 포함되어야 한다.

최적 이동 패턴의 탐사 과정을 보이기 위해, (표 1)은 (그림 1)의 도로 네트워크 상에서 각 경로 구간을 이동할 때 소요되는 비용, 즉 가중치 값을 나타낸 것이고, (표 2)는 이동 객체들의 궤적을 각 노드에서 샘플링 한 이력 데이터이다. 가중치는 이동 객체의 각 이동 정보들 간의 효율성을 상대적으로 비교하여 백분율로 표현한 것으로, 가중치의 기준요소는 이동시간 또는 이동거리, 이동 소요 비용 등이 될 수 있다.



(그림 1) 공간지역과 도로 네트워크 예

(표 1) 이동 경로별 가중치 예

Path	weight	Path	weight	Path	weight	Path	weight
P1	0.76	P2	0.12	P3	0.24	P4	0.18
P5	0.48	P6	0.28	P7	0.43	P8	0.56
P9	0.71	P10	0.61	P11	0.42	P12	0.54
P13	0.26	P14	0.64	P15	0.55	P16	0.74
P17	0.15	P18	0.72	P19	0.64	P20	0.85
P21	0.42	P22	0.68	P23	0.45	P24	0.45
P25	0.24	P26	0.34	P27	0.24	P28	0.77
P29	0.24	P30	0.58	P31	0.33	P32	0.61
P33	0.45	P34	0.51	P35	0.16	P36	0.14
P37	0.79	P38	0.37	P39	0.15	P40	0.44
P41	0.87	P42	0.55	P40	0.44	P44	0.47

(표 2) 이동 객체의 이력 데이터 예

OID	X	Y	VT	OID	X	Y	VT
1	84	166	2005/11/20/12/10/08	6	84	166	2005/11/20/12/10/24
	108	146	2005/11/20/12/21/19		108	146	2005/11/20/12/27/15
	121	133	2005/11/20/12/29/34		88	125	2005/11/20/12/36/34
	95	99	2005/11/20/12/43/36		95	99	2005/11/20/12/50/45
	139	108	2005/11/20/12/56/16		139	108	2005/11/20/13/02/12
	157	90	2005/11/20/13/04/25		157	90	2005/11/20/13/10/54
	191	81	2005/11/20/13/11/16		191	81	2005/11/20/13/22/24
	226	66	2005/11/20/13/15/16		226	66	2005/11/20/13/31/14
2	88	125	2005/11/20/12/02/55	7	78	90	2005/11/20/12/37/43
	95	99	2005/11/20/12/09/41		100	70	2005/11/20/12/45/39
	116	87	2005/11/20/12/15/15		116	87	2005/11/20/12/53/26
	139	72	2005/11/20/12/21/13		139	72	2005/11/20/12/59/34
	157	90	2005/11/20/12/26/15		157	90	2005/11/20/13/16/14
	191	81	2005/11/20/12/33/34		191	81	2005/11/20/13/27/12
	226	66	2005/11/20/12/41/45		226	66	2005/11/20/13/36/32
3	84	166	2005/11/20/12/33/35	8	116	87	2005/11/20/12/14/35
	108	146	2005/11/20/12/54/14		100	70	2005/11/20/12/23/54
	121	133	2005/11/20/13/15/11		119	52	2005/11/20/12/33/27
	139	108	2005/11/20/13/29/12		134	51	2005/11/20/12/39/19
	157	90	2005/11/20/13/43/36		163	48	2005/11/20/12/48/35
	191	81	2005/11/20/13/52/29		191	81	2005/11/20/13/00/26
	226	66	2005/11/20/13/01/19		226	66	2005/11/20/13/07/07
	4	95	99		2005/11/20/12/10/05	9	150
139		108	2005/11/20/12/20/54	156	160		2005/11/20/12/34/24
157		90	2005/11/20/12/28/29	148	134		2005/11/20/12/42/34
157		53	2005/11/20/12/39/38	173	108		2005/11/20/12/49/25
163		48	2005/11/20/13/00/27	199	108		2005/11/20/12/56/43
191		81	2005/11/20/13/12/06	191	81		2005/11/20/13/05/16
226		66	2005/11/20/13/20/46	226	66		2005/11/20/13/12/15
5	121	133	2005/11/20/12/15/45	10	195	166	2005/11/20/12/15/04
	95	99	2005/11/20/12/22/54		182	151	2005/11/20/12/20/43
	139	108	2005/11/20/12/34/07		173	108	2005/11/20/12/29/48
	157	90	2005/11/20/12/45/46		199	108	2005/11/20/12/34/23
	191	81	2005/11/20/12/55/15		191	81	2005/11/20/13/20/26
	226	66	2005/11/20/13/04/33		226	66	2005/11/20/13/26/00

3.2 최적 이동 패턴 탐사 과정

최적 이동 패턴의 탐사 과정은 이동 객체의 이력 데이터를 변환하는 전처리 단계와 최적 이동 패턴을 탐사하는 단계로 구분된다. 전처리 단계는 객체의 이력 데이터로부터 이동 시퀀스를 생성하는 단계로, 빈발도 기반의 최적 이동 패턴 탐사 방법[25]에서의 이동 시퀀스 생성 방법과 동일하다. 이동 시퀀스란 이동 객체의 이력 데이터를 각 객체별로 시간 순차에 따라 나열한 한 쌍의 공간

및 시간 속성값으로 구성된 순차 리스트로서, 이동 객체 데이터베이스를 각 객체별로 유효시간을 기준으로 정렬한 후 시퀀스를 구성하는 위치들 간에 최대 시간 간격 max_gap 에 대한 제약 조건을 두어 이를 만족할 때 이동 시퀀스를 생성한다. 다음 (표 3)은 (표 2)의 이력 데이터로부터 시간 간격을 고려하여 이동 시퀀스를 추출한 후 시퀀스 데이터의 공간 속성을 노드 식별자로 대치하여 생성한 이동 시퀀스 집합이다.

(표 3) 최적 이동 패턴 탐사를 위한 이동 시퀀스 집합 예

OID	Sequence
MO1	<N1 N3 N15 N17 N14 N19 N20 N21>
MO2	<N16 N17 N22 N23 N19 N20 N21>
MO3	<N1 N3 N15 N14 N19 N20 N21>
MO4	<N17 N14 N19 N24> <N25 N20 N21>
MO5	<N15 N17 N14 N19 N20 N21>
MO6	<N1 N3 N16 N17 N14 N19 N20 N21>
MO7	<N30 N29 N22 N23> <N19 N20 N21>
MO8	<N22 N29 N28 N26 N25 N20 N21>
MO9	<N6 N7 N13 N12 N10 N20 N21>
MO10	<N3 N15 N14 N19> <N20 N21>

최적 이동 패턴 탐사 단계는 전처리 과정을 통해 생성된 이동 시퀀스 집합으로부터 최소지지도 min_sup 이상의 지지도를 가진 빈발 2-시퀀스들 중 가장 최소의 가중치를 가지는 이동 시퀀스들을 탐사하는 단계이다. 패턴 탐사의 기본 원리는 먼저, 트랜잭션 데이터베이스로부터 시간 및 공간 제약을 만족하는 이동 시퀀스 집합을 생성한다. 그 다음 공간상의 특정한 두 지점(시작점, 종료점)을 항목으로 가지는 후보 시퀀스 집합을 생성하고, 각 후보 시퀀스들로부터 첫 번째와 두 번째 항목만을 추출하여 부분 2-시퀀스를 생성한다. 생성된 부분 2-시퀀스들에 대해 이동 시퀀스 집합의 각 시퀀스들을 탐색하여 지지도를 계산한 후 지지도가 min_sup 이상인 빈발 2-시퀀스들을 추출하고 이 중 가장 적은 가중치를 갖는 최소 가중 2-시퀀스를 생성한다. 이러한 과정은 최소 가중 2-시퀀스의 항목으로 종료점 F가 포함된 2-시퀀스를 추출할 때까지 반복적으로 수행되며, 모든 최소 가중 2-시퀀스들을 탐색했을 경우 각 시퀀스를 구성하는 항목들의 순서에 따라 조합하여 최적 이동 패턴을 추출한다.

[단계 1] 후보 시퀀스 집합 생성 단계

이동 시퀀스 집합으로부터 각 시퀀스를 구성하는 항목들 중 첫 번째 항목이 S이고 마지막 항목이 F인 모든 후보 시퀀스 집합을 추출한 후 S 항목과 이에 직접 연결된 항목으로 구성되는 2-시퀀스 집합을 추출한다. 최초의 시작점은 S이고 다음 패스에서의 시작점 S'은 [단계 2]에서 추출한 최소 가중 2-시퀀스의 두 번째 항목이다.

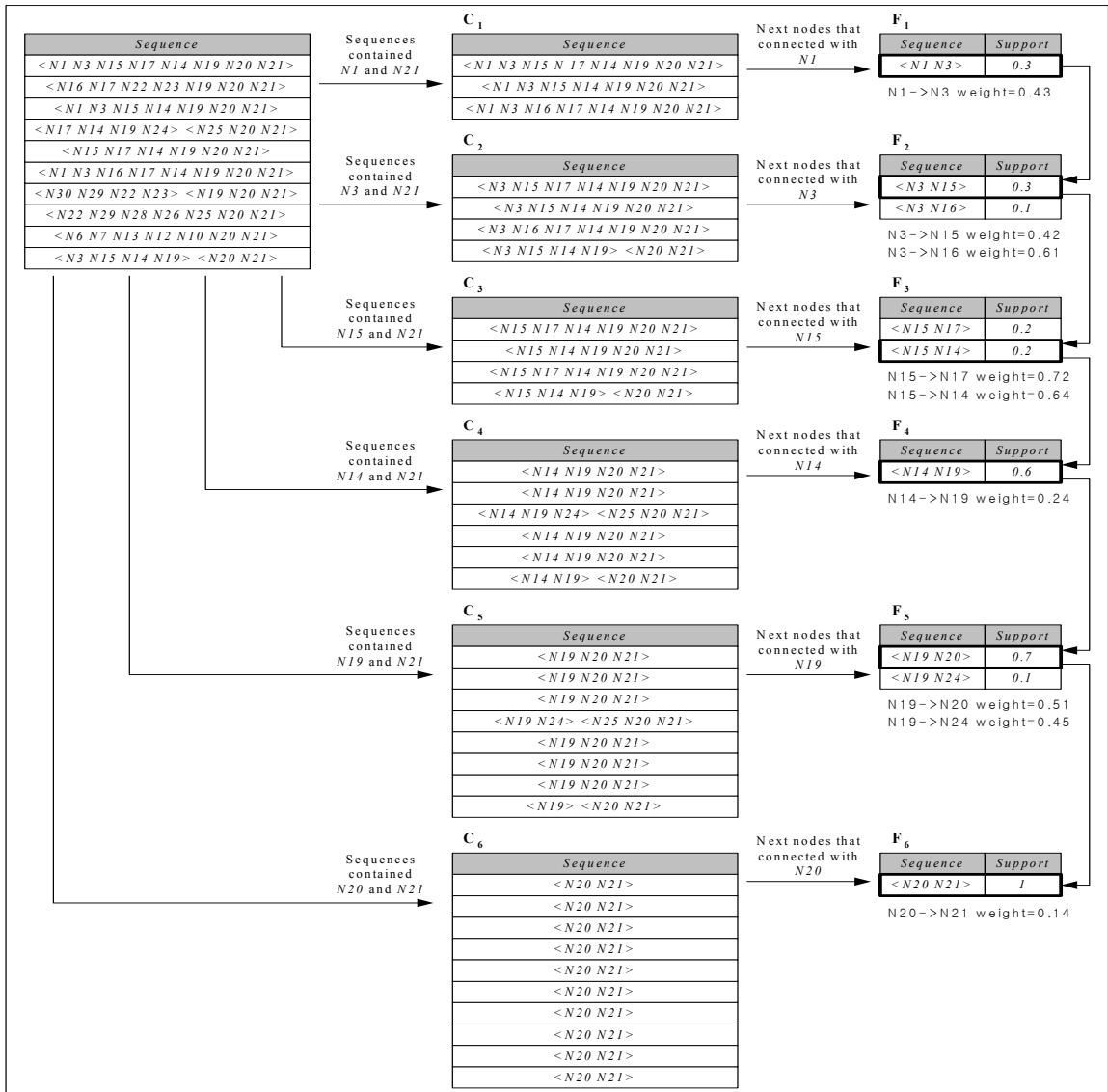
[단계 2] 최소 가중 2-시퀀스 추출 단계

추출한 2-시퀀스 집합에 대해 지지도를 계산하여 최소 지지도 min_sup 이상을 갖는 빈발 2-시퀀스들만 추출한 후 이들 중 가장 작은 가중치를 갖는 최소 가중 2-시퀀스를 추출한다. 추출된 최소 가중 2-시퀀스를 구성하는 두 항목 중 두 번째 항목이 종료점 F와 같지 않으면 [단계 1]을 반복하고 같으면 [단계 3]을 수행한다.

[단계 3] 최적 이동 패턴 탐사 단계

[단계 1]과 [단계 2]의 과정을 거쳐 생성된 최소 가중 2-시퀀스들을 순차적으로 연결하여 최적 이동 패턴을 추출한다.

이와 같은 탐사 과정은 빈발도 기반의 최적 이동 패턴 탐사 방법[25]과 거의 유사하지만 탐사과정 중 마지막 최대 빈발 2-시퀀스를 결정하는 단계에서 빈발도 기반은 최대지지도를 기준으로, 빈발도와 가중치 기반은 최소가중치를 기준으로 결정한다는 점에서 서로 다르다. 빈발도와 가중치를 기준으로 (표 3)의 이동 시퀀스 집합을 이용하여 최적 이동 패턴을 탐사하는 과정을 보이면 (그림 2)와 같다. (그림 2)의 탐사 과정에서는 빈발도가 사용자에게 의해 지정된 최소지지도 min_sup 보다 크거나 같은 항목들에 대해 가중치를 검사하여 가장 최소의 가중치를 가지는 부분 2-시퀀스를 선택한다. 가령, 최소지지도가 $min_sup \leq 0.2$ 로 지정되었을 경우, 두 번째 패스에서는 F2 집합의 두 시퀀스 중 하나만 min_sup 을 만족하므로 <N3 N15> 시퀀스가 결정되고, 세 번째 패스에서는 F3 집합의 두 시퀀스 모두 동일한 지지도를 가짐으로서 두 시퀀스에 대한 가중치를 비교하여 가중치가 작은 <N15 N14>를 최소 가중 2-시퀀스로 선택한다. 이러한 탐사 과정을 반복적으로 수행하면 최종적으로 <N1 N3 N15 N14 N19 N20 N21>의 최적 이동 패턴을 추출할 수 있다.



(그림 2) 빈발도와 가중치 기반의 최적 이동 패턴 탐사 예

3.3 최적 이동 패턴 탐사 알고리즘

빈발도와 가중치 기반의 최적 이동 패턴 탐사 알고리즘 *STOMP(FW)*는 선행연구에서 제안했던 빈발도 기반의 패턴 탐사 알고리즘 *STOMP(F)*[25]와 거의 유사한 형태로 설계될 수 있다. 하지만, 최적 패턴을 탐사하는데 있어 이동 패턴의 빈발

도에 대한 임계치인 최소지지도 *min_sup*과 각 단위 경로에 대한 가중치 *WT*를 사용한다는 점에서 다르다. (그림 3)의 *STOMP(FW)*는 *STOMP(F)*와 마찬가지로 *SeqExtractor* 함수를 통해 이동 시퀀스를 생성한 후 *OptPathExtractor-WT* 함수를 통해 최대 빈발 시퀀스를 추출하여 최적 경로를 결정한다. *limitedDataSet* 함수는 이동 객체 데이터베이스

스로부터 공간 및 시간 한정자 C_s 와 C_t 공간제 약조건 S 와 F , 두 제약 조건 모두를 만족하는 객체의 이력 데이터 집합을 추출하는 함수이다.

- (1) 최적 경로 추출 알고리즘 :
OptPathExtractor-WT

OptPathExtractor-WT 함수는 최적 이동 패턴을 탐색하기 위해서 반복적으로 *Freq_Link-WT* 함수를 호출하여 출발점 S 로부터 시작해 도착점 F 에 도달할 때까지의 단위 최적 패턴을 탐사한다. 반복적인 단위 최적 패턴 탐사가 종료되면, *OptPath*의 최종 결과 값이 최적 이동 패턴으로 결정된다.

Input : D (Database), C_s (Spatial Constraint), C_t (Temporal Constraint),
 z S (Start Node), F (Final Node), max_gap (Time Interval Constraint),
 Ls_k (Spatial Level), min_sup (Thershold of Minimum Support),
 WT (Threshold of Weight)

Output : *OptPath*

Procedure $STOMP^{FW}(D, C_s, C_t, S, F, max_gap, Ls_k, min_sup, WT)$

Begin

$PrevA = null;$

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

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

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

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

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

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

//Moving Pattern Minging

$OptPath = OptPathExtractor - WT(D_{Rgen}, min_sup, WT, S, F);$

Return $OptPath;$

End

(그림 3) $STOMP(FW)$ 알고리즘

Input : D_{Rgen} (Generalized Moving Sequence), S (Start Node), F (Final Node),
 min_sup (Threshold of Minimum Support), WT (Threshold of Weight)

Output : $OptPath$

Procedure $OptPathExtractor - WT (D_{Rgen}, min_sup, WT, S, F)$

Begin

$PrevA = null;$

$OptPath = null;$

$nextN = S;$

While ($nextN \neq F$) {

$OptPath$ add $nextN;$

$nextN = Freq_Link - WT(nextN, F, D_{Rgen}, min_sup, WT);$

}

$OptPath$ add $F;$

Return $OptPath;$

End

(그림 4) $OptPathExtractor-WT$ 알고리즘

(2) 단위 최적 패턴 추출 알고리즘 :

$Freq_Link-WT$

단위 최적 패턴을 추출하기 위한 $Freq_Link-WT$ 함수는 $OptPathExtractor-WT$ 함수에서 전달받은 전체 이동 시퀀스 집합으로부터 출발점 S 와 도착점 F 를 포함하는 부분 이동 시퀀스 집합을 추출한다. 그 다음 최초 S 로부터 시작하는 최소지지도 min_sup 이상의 1-빈발 패턴들을 탐색하고 그 중 가중치가 최소인 패턴을 추출한다. 추출된 패턴에서 S 가 아닌 반대편 노드 S' ($nextN$)을 기준으로 하여 다시 F 를 포함하는 모든 부분 이동 시퀀스 집합을 추출한 후 최소지지도 및 가중치 조건을 만족하는 새로운 단위 최적 패턴을 추출한다. 이러

한 과정은 F 에 도달하는 단위 최적 패턴을 탐사할 때까지 반복 수행한다.

4. 실험 및 성능 평가

본 장에서는 최적 이동 패턴을 추출하기 위한 $STOMP(FW)$ 알고리즘의 성능을 평가한다. 제안된 $STOMP(FW)$ 알고리즘에 대한 성능 평가는 가장 일반적인 최적 경로 탐색 알고리즘인 $Dijkstra$ 와 A^* 알고리즘들을 대상으로 실험을 통해 성능을 비교, 분석하였다. 단, A^* 알고리즘의 경우는 휴리스틱 가중치를 한 범위만 지정할 수 없어 10~20Km 구간과 20~30Km 구간으로 나누어 실험하였다. 실험 조건은 모두 동일하며 각 100, 200,

Input : $currN$ (Current Node), F (Final Node), D_{Rgen} (Generalized Moving Sequence),
 min_sup (Threshold of Minimum Support), WT (Threshold of Weight)

Output : Node

Procedure $Freq_Link - WT(currN, F, D_{Rgen}, min_sup, WT)$

Begin

$subD = Search\ Sequence\ that\ contained\ Node\ currN\ and\ Node\ F\ in\ D_{Rgen};$

For each $moving_sequence\ S \in subD$

$c = subset(currN, S); //Link\ from\ currN\ to\ nextN\ that\ connected\ with\ currN$

If $c \notin C_s$ *Then* $//Set\ of\ Links\ between\ the\ decided\ Node\ and\ optional\ Node$

C_s *add* $c;$

End If

$c.count++;$

End For

$Freq_{seq} = find\ 2 - sequence\ with\ support\ great\ than\ min_sup\ from\ C_s;$

$c_{min_weight} = find\ 2 - sequence\ with\ threshold\ great\ than\ WT\ from\ Freq_{seq};$

$nextN = another\ Node\ that\ connected\ with\ currN\ in\ c_{min_weight};$

Return $nextN;$

End

(그림 4) $Freq_Link-WT$ 알고리즘

300, 400, 500번의 경로 탐색을 통해 평균 데이터를 산출하였다. 경로 탐색을 위해 서울시를 탐색 범위로 설정하고, 각 탐색 횟수(100~500번)에 대비하여 출발점(서울시 중구내 임의노드)을 중심으로 동심원 형태의 5개 레벨로 반경을 점차 늘려 가며 해당 반경에 포함되는 도착점까지의 거리에 대한 결과를 측정하였다. STOMP(FW)의 최소지지도는 STOMP(F)가 최대지지도를 갖는 최적 패턴을 탐색했을 때, 각 구간의 최대지지도 중 가장 작은 지지도를 최소지지도로 적용하였다. 최적 경로 탐색 알고리즘들에 대한 성능 평가는 연산 처리 시간과 최적경로 정확도를 기준으로 각 알고리즘들의 결과값을 추출하였다.

실험을 위한 시공간 이동 패턴 탐사 시스템

STMPMiner의 개발을 위해 Windows XP 기반하에서 Eclipse 3.2와 JDK 1.6.0을 사용하였고, 이동 객체 데이터와 Geometry 데이터를 저장하기 위해 Oracle 10g로 데이터베이스를 구축하였다. 성능 실험 환경은 STMPMiner 시스템과 데이터베이스 시스템으로 각각 Pentium IV 2.4GHz, 2GByte Memory PC와 Pentium IV 2.0GHz, 1GByte Memory PC를 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. 이력 데이터는 도로 네트워크 상에서의 교차점이나 분기점, 종료점 등을 노드로 설정하여 각 노드에서

이동 객체의 위치 정보를 샘플링하였다. 다음 (표 4)는 성능 평가 실험에 이용되는 이력 데이터의 유형이고, (표 5)는 실험 데이터에 대한 특성을 나타낸다. 데이터 집합의 이름에서 N은 이동 객체 수, D는 이동기간, G는 샘플링 위치를 의미한다.

(표 4) 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 5) 실험 데이터 특성

데이터 집합 이름	이동 객체 수 (N)	이동기간 (D)	실험 데이터 크기
N4000-D1014-Gnode	4000	1일	58.9 MB

4.1 평균 경로 탐색 연산 시간의 성능 평가

경로 탐색에 있어서의 평균 연산 시간은 (표 6)과 같이 STOMP(FW), STOMP(F), Dijkstra, A* 알고리즘들에 대해 연산 처리 횟수 100, 200, 300, 400, 500회 동안 동일한 공간 범위의 경로들을 탐색하도록 하여 연산 처리 시간을 산출하였다. 평균 연산 시간은 각 탐색 수행 횟수만큼 특정 출발점에서 동일한 레벨 내에 있는 도착점을 임의로 지정하고, 각 알고리즘의 연산 시간 산정 방식에 따라 연산시간들을 추출하여 평균을 구한 시간이다.

(표 6) 평균 경로 탐색 연산 시간 실험 결과

수행 횟수	STOMP (FW)	STOMP (F)	A* (10~20Km)	A* (20~30Km)	Dijkstra
100	1.45	0.88	0.32	5.07	36.34
200	1.56	0.93	0.41	5.78	64.54
300	1.59	1.01	0.38	5.34	145.36
400	1.64	1.22	0.39	4.96	324.32
500	1.71	1.26	0.42	5.98	724.34

실험 결과, 각 탐색 수행횟수에 대응하여 출발점(서울시 중구내 임의노드)을 중심으로 동심원 형태의 5개 레벨로 반경을 확대하기 때문에 출발점으로부터 도착점까지의 거리가 점차 멀어짐으로써 평균 연산 시간이 증가하였다. A* 알고리즘의 휴리스틱 가중치를 10~20Km 구간으로 지정하였을 때 가장 좋은 연산 처리 성능을 보였으며, STOMP(FW), A*(20~30Km), Dijkstra 알고리즘 순으로 연산 처리 성능이 다소 차이를 보였다. 하지만 A*(10~20Km) 알고리즘의 경우 연산 처리 시간이 짧은 장점이 있지만 목적지 방향으로의 직진성이 강해 최적 경로를 찾지 못할 확률이 매우 높았다. 또한 A* 알고리즘의 경우 휴리스틱 가중치 범위에 따라 연산 처리 시간이 다소 큰 차이를 보이는데, 목적지 방향으로의 직진성 범위에 따라 탐색 대상 노드 수가 증가하여 연산 처리 시간이 늘어났기 때문이다. Dijkstra 알고리즘은 출발점을 중심으로 동심원 형태로 모든 노드를 대상으로 하여 경로 탐색을 수행하기 때문에 가장 많은 연산 처리 시간을 소요하게 된다. STOMP(FW) 알고리즘은 빈발도 및 가중치 기반의 탐사 과정을 통해 최적 경로를 추출하기 때문에 기존의 STOMP(F) 알고리즘 보다 약간의 연산 시간 이 더 요구되어 위와 같은 결과를 보였다.

4.2 최적 경로의 정확도에 대한 성능 평가

Dijkstra 알고리즘은 지도상의 모든 노드를 대상으로 하여 탐색을 수행하기 때문에 어떠한 경우라도 최적 경로를 탐색할 수 있다. 따라서 Dijkstra 알고리즘에 의해 탐색된 최적 경로의 정확도는 100%이다. 최적 경로의 정확도는 이러한 Dijkstra 알고리즘에 의해 탐색된 최적 경로와 여러 유형의 최적 경로 탐색 알고리즘에 의해 탐색된 최적 경로들을 비교하여 얼마만큼 근접했는지를 평가하는 기준이다. 다음 (표 7)은 STOMP(FW)와 STOMP(F), Dijkstra, A* 알고리즘들에 대해 연산 처리 횟수 100, 200, 300, 400, 500회 동안 동일한 공간 범위에서 최적 경로를 탐색했을 때 실제

의 최적 경로(Dijkstra의 최적 경로)와 정확도를 비교한 결과이다.

(표 7) 최적 경로 정확도 실험 결과

수행 횟수	STOMP (FW)	STOMP (F)	A* (10~20Km)	A* (20~30Km)	Dijkstra
100	94.25	88.25	78.36	90.14	100.00
200	95.92	87.92	81.34	87.32	100.00
300	96.64	89.64	76.31	88.00	100.00
400	95.34	88.34	83.74	89.31	100.00
500	94.54	90.54	82.64	92.14	100.00

(표 7)의 결과에서 STOMP(FW) 알고리즘에 의해 탐색되는 경로들의 정확도가 Dijkstra 알고리즘의 최적 경로 정확도에 가장 근접하는 것을 확인할 수 있다. 이는 빈발도가 높은 패턴들을 대상으로 최적 경로를 탐색하기 때문에 목표지점에 도달하는 경로를 반드시 찾을 수 있으며, 최소의 가중치를 고려함으로써 최적 경로에 가깝게 접근할 수 있었다. A* 알고리즘의 경우 휴리스틱 가중치가 10~20Km 구간으로 설정되었을 때 최적 경로의 정확도가 가장 떨어지는데, 이는 휴리스틱 가중치에 의해 직진성이 강해져 탐색 범위가 작아지기 때문이다. 이러한 이유로 휴리스틱 가중치를 넓게 지정하여 많은 노드들을 탐색함으로써 보다 정확한 최적 경로를 찾아낼 수는 있지만, 앞서 실험한 결과를 보면 접근 노드의 수가 많을 경우 연산시간이 증가하는 문제가 있기 때문에 적절한 휴리스틱 가중치의 설정이 중요하다.

5. 결론 및 향후 연구과제

실세계의 응용 분야에서 활용할 수 있는 새로운 위치 기반 서비스를 개발하기 위한 목적으로 다양한 시공간 이동 패턴 탐사 기법에 대한 연구가 진행되고 있다. 하지만 기존에 연구된 탐사 기법들은 시간 및 공간 한정자에 의해 제한되어진 시간 및 공간영역에서의 모든 이동 패턴들 중 불특정한 빈발 이동 패턴들만을 탐사하기 때문에

사용자 요구에 맞는 보다 정확한 패턴을 추출하는데 한계가 있다. 이에 본 논문에서는 이동 객체의 이력 데이터 집합으로부터 생성된 다양한 이동 패턴들 중 복합적인 시간 및 공간 제약 조건을 만족하고, 실세계에서의 최적 경로에 해당하는 최적 이동 패턴을 탐사하기 위한 STOMP(FW) 기법을 제시하였다. 제안된 기법에서는 3.1절의 [가정 1,2,3]에 따라 다양한 이동 패턴들 중 최소지지도 이상의 빈발도를 가지며, 이동시간이나 이동거리, 소요비용 등과 같은 가중치가 최소인 이동 패턴을 탐사한다. 이러한 STOMP(FW) 알고리즘의 구성은 SeqExtractor 연산을 통해 이동 객체의 이력 데이터를 시퀀스로 변환시키고, Contains 연산을 통해 이동 객체의 위치 속성을 공간영역으로 일반화하여 보다 효율적인 패턴 탐사가 이루어질 수 있도록 설계 및 구현하였다. 또한 OptPathExtractor-WT와 Freq_link-WT 연산을 통해 빈발도와 가중치를 모두 고려하여 최적 이동 패턴을 추출할 수 있도록 각 알고리즘들을 설계 및 구현하였다.

STOMP(FW) 알고리즘에 대한 성능 실험은 최적 경로 탐색에 일반적으로 가장 많이 사용되는 Dijkstra와 A* 알고리즘을 대상으로 연산 처리 시간과 최적경로 정확도를 기준으로 성능을 비교, 평가하였다. A* 알고리즘의 경우는 휴리스틱 가중치에 따라 실험 결과 값이 큰 차이를 보이기 때문에 10~20Km 구간과 20~30Km 구간으로 나누어 실험하였다. 먼저 연산 처리 시간을 기준으로 실험한 결과, 가장 좋은 연산 처리 성능을 보인 A*(10~20Km) 알고리즘의 경우, 최적 경로를 탐색하지 못할 확률이 높은 반면 대체로 안정적으로 최적 경로를 탐색할 확률이 높은 A*(20~30Km) 알고리즘의 경우, 연산 처리 시간이 STOMP(FW) 알고리즘보다 비효율적이었다. 이는 이동 객체가 과거에 이동했던 경험 데이터로부터 빈발도가 특정 임계치 이상인 패턴들을 추출한 후 가중치를 고려해 최적 경로를 탐색하기 때문에 접근 노드 수가 A* 알고리즘에 비해 적어 연

산 처리 시간이 적게 소요되었다. 최적 경로의 정확도에 대한 실험 결과는 Dijkstra 알고리즘에 의해 탐색된 경로가 정확도가 100%임으로, 이를 기준으로 제안된 STOMP(FW)와 A* 알고리즘들을 비교한 결과 STOMP(FW) 알고리즘이 A* 알고리즘에 비해 좋은 정확도를 보였다. 또한, 패턴 빈발도만을 이용한 기존의 STOMP(F) 알고리즘보다 가중치 적용을 위한 연산 시간은 약간 증가하지만 최적 경로 정확도는 연산 처리 횟수에 따라 5% ~ 8%까지 증가함을 알 수 있다.

STOMP(FW) 방법은 기존의 최적 경로 탐색 방법(STOMP(F), A*, Dijkstra)과 비교하여 패턴 탐사 시간을 최소화하고 최적 패턴의 정확도는 극대화하기 위한 방식이다. 제안된 방식에 의해 추출된 최적 패턴은 물류 관계 분야에서의 최적 물품 경로 서비스나 최적 경로 추천 서비스 등으로 제공될 수 있는데, 이러한 서비스들은 짧은 시간 내에 최적 경로를 제공하는 것도 중요하지만 가장 정확한 최적 경로를 제공함으로써 이동 시 소요되는 비용을 최소화하는 데 목적이 있다. 따라서 STOMP(FW) 방식은 기존의 빈발도만 이용하는 방식에서 가중치까지 고려하여 패턴 탐사를 수행함으로써 최적 이동 패턴의 정확도를 높여 이동 소요 비용을 최소화할 수 있다는 점에서 의미가 크다.

또한, 본 논문에서 사용한 이력 데이터는 실제 계의 경험 이력 데이터를 기반으로 하기 때문에 특정 경로 상에서의 교통 체증이나 도로공사와 같은 특수한 상황에 대한 이력도 포함할 수 있다. 가령 특정 구간에서 교통 체증이나 도로공사가 발생했을 경우, 기존의 최적 경로 탐색 방법들은 이러한 상황을 경로 탐색에 반영하지 못하여 문제의 구간을 최적 경로에 포함시킬 수 있지만, STOMP(F)와 STOMP(FW) 방식은 빈발도를 이용하여 패턴 탐사를 수행하기 때문에 이러한 구간을 회피하는 이동 패턴들을 탐사할 수 있어 보다 능동적으로 최적 경로를 탐사할 수 있다.

향후 연구과제로는 최적 이동 패턴 탐사 기법

을 이용하여 단위 시간동안 이동 객체가 순회해야 하는 지점들에 대한 스케줄링 경로 예측을 위한 마이닝 기법의 개발이 필요하다.

참 고 문 헌

- [1] A. P. Sistla, O. Wolfson, S. Chamberlain and S. Dao, "Querying the Uncertain Position of Moving Objects", Temporal Databases: Research and Practice, Springer Verlag Lecture Notes in Computer Science number 1399, 1998.
- [2] O. Wolfson, S. Chamberlain, S. Dao, L. Jiang and G. Mendez, "Cost and Imprecision in Modeling the Position of Moving Objects", in proc. on the 14th International Conference on Data Engineering, 1998.
- [3] O. Wolfson, A. P. Sistla, B. Xu, J. Zhou and S. Chamberlain, "DOMINO : Databases fOr MovINg Objects tracking", in proc. ACM SIGMOD Symp. on the Management of Data, 1999.
- [4] O. Wolfson, B. Xu, J. Zhou, S. Chamberlain, Y. Yesha and N. Rishe, "Tracking Moving Objects Using Database Technology in DOMINO", in proc. on The Fourth Workshop on Next Generation Information Technologies and Systems(NGITS), pp.112-119, July 1999.
- [5] 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.
- [6] N. Mamoulis, H. Cao, G. Kollis, 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.

- [7] 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.
- [8] Y. Huang, L. Zhang and P. Zhang, "Finding Sequential Patterns from a Massive Number of Spatio-Temporal Events", SDM, SIAM, 2006.
- [9] 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.
- [10] 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.
- [11] S. Y. Hawng, Y. H. Liu, J. K. Chiu and E. P. Lim, "Mining Mobile Group Patterns : A Trajectory-based Approach", PAKDD, Lecture Notes in Computer Science, Springer, Vol.3518, pp.713-718, 2005.
- [12] Y. Wnag, E. P. Lim and S. Y. Hwang, "On Mining Group Patterns of Mobile Users", DEXA, Lecture Notes in Computer Science, Springer, Vol.2736, pp.287-296, 2003.
- [13] J. Gudmundsson, M. V. Kreveld and B. Speckmann, "Efficient Detection of Motion Patterns in Spatio-Temporal Data Sets", in proc. on the 12th annual ACM international workshop on Geographic Information Systems(GIS), pp.250-257, 2004.
- [14] P. Laube and S. Imfeld, "Analyzing Relative Motion within Groups of Trackable Moving Point Object", in GIScience, Notes in Computer Science, Springer, Vol.2478, pp.132-144, 2002.
- [15] 김도현, 진희채, "텔레메틱스와 위치 기반 서비스를 위한 차세대 통신 기술", 한국멀티미디어 학회지, 제7권, 제4호, pp.40-51. 2003.
- [16] 최희옥, 김남진, "웹 서비스를 이용한 텔레메틱스 기술의 응용", 한국정보처리학회지 논문지, 제9권, 제4호, 2002.
- [17] 유한경, "SMS와 GPS를 활용한 실시간 물류상황 관제시스템의 개발", 인천대학교 대학원, 석사학위논문, 2004.
- [18] 안윤애, 김동호, 류근호, "차량위치 추적을 위한 이동객체 관리 시스템의 설계", 한국정보처리학회 논문지, 제9-D권, 제5호, 2002.
- [19] 안윤애, 류근호, "이동 객체의 불확실한 과거 및 미래의 위치 추정", 한국정보처리학회 논문지: 데이터베이스, 제29권, 제6호, 2002.
- [20] 안윤애, 류근호, "이동 객체의 과거 및 미래의 위치 연산을 위한 데이터 모델", 한국정보처리학회 논문지, 제10-D권, 제1호, 2003.
- [21] 백옥현, "위치 기반 서비스를 위한 이동 객체의 시간 패턴 탐사 기법", 충북대학교 대학원, 석사학위논문, 2002.
- [22] 이준욱, "위치 기반 서비스를 위한 이동 객체의 시간 패턴 탐사", 한국정보과학회 논문지, 제29권, 제5호, 2002.
- [23] 이준욱, "지식 탐사 프레임워크 기반의 시공간 이동 패턴 탐사 기법", 충북대학교 대학원, 박사학위논문, 2003.
- [24] 한선영, "시공간 이동 시퀀스 패턴 마이닝 기법", 이화여자대학교 대학원, 석사학위논문, 2006.
- [25] 고 현, 이연식, "이동 시퀀스의 빈발도를 이용한 최적 이동 패턴 탐사 기법", 한국정보처리학회 논문지, 제16-D권, 제1호, 2009.

● 저 자 소개 ●



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



박 성 숙

1992년 원광대학교 컴퓨터공학과(학사)
1995년 원광대학교 대학원 컴퓨터공학과(공학석사)
2007년 군산대학교 대학원 컴퓨터정보공학과 박사과정
1996년~2006년 홍성기능대학 교수
2003년~2006년 홍성기능대학 전산실 실장
2007년~현재 한국폴리텍 V 김제 대학 교수
2007년~현재 한국폴리텍 V 김제 대학 전산실 실장
관심분야: 데이터 마이닝, 지능형 에이전트, 센서 네트워크 미들웨어
e-mail : joara@kopo.ac.kr