

도로 네트워크에서 이동 객체를 위한 시공간 유사 궤적 검색 알고리즘[†]

Trajectory Search Algorithm for Spatio-temporal Similarity of Moving Objects on Road Network

김영창* / Young-Chang Kim 라빈드라 비스타** / Rabindra Vista 장재우*** / Jae-Woo Chang

요약

모바일 환경의 대중화와 이를 위한 기반 기술의 발전으로 인하여 이동 객체들을 효과적으로 표현하고 분석하는 것이 중요한 문제로 대두되고 있다. 이러한 환경에서 이동 객체 궤적의 유사성 검색은 궤적에 대한 데이터 마이닝의 일부분으로 중요한 연구 분야중의 하나이다. 본 논문에서는 도로 네트워크 상의 이동 객체 궤적을 위한 시공간 유사 궤적 검색 알고리즘을 제안한다. 이를 위하여 도로 네트워크 상에서 두 이동 객체 궤적 사이의 시공간 거리를 정의하고, 이를 기반으로 궤적 사이의 시공간 유사도 측정 방법을 제안한다. 유사 궤적 알고리즘은 효율적인 검색을 위하여 시그니처 파일 기법을 이용하여 궤적을 검색한다. 마지막으로, 본 논문에서 제안하는 시공간 유사 궤적 검색 알고리즘을 구현하고, 성능 분석을 통해 제안하는 알고리즘의 효율성을 입증한다.

Abstract

Advances in mobile techknowledges and supporting techniques require an effective representation and analysis of moving objects. Similarity search of moving object trajectories is an active research area in data mining. In this paper, we propose a trajectory search algorithm for spatio-temporal similarity of moving objects on road network. For this, we define spatio-temporal distance between two trajectories of moving objects on road networks, and propose a new method to measure spatio-temporal similarity based on the real road network distance. In addition, we propose a similar trajectory search algorithm that retrieves spatio-temporal similar trajectories in the road network. The algorithm uses a signature file in order to retrieve candidate trajectories efficiently. Finally, we provide performance analysis to show the efficiency of the proposed algorithm.

주요어 : 유사 궤적 검색, 이동 객체, 도로 네트워크 데이터베이스

Keyword : Similar trajectory search, moving objects, road network database

† 이 논문은 교육인적자원부, 산업자원부, 노동부의 출연금으로 수행한 최우수실험실지원사업의 연구결과입니다.

■ 논문접수 : 2007.3.12 ■ 심사완료 : 2007.6.5

* 교신저자 전북대학교 전자정보공학부 컴퓨터공학 박사과정 (yckim@dblab.chonbuk.ac.kr)

** 전북대학교 전자정보공학부 컴퓨터공학 박사과정 (rabindra@dblab.chonbuk.ac.kr)

*** 전북대학교 전자정보공학부 컴퓨터공학 교수 (jwchang@chonbuk.ac.kr)

1. 서론

이동 객체를 위한 유사 궤적 검색은 많은 분야에서 관심을 갖는 연구주제이다. 궤적 데이터는 기상학, 무선응용, 영상추적 [1], 모션 캡처 [2] 등을 포함한 많은 분야에서의 공통적으로 다루어지는 데이터이다. 최근 휴대용 전자 장치, 센서 네트워크, GPS 기술의 발전으로 대용량의 시공간 데이터의 수집이 가능하게 되었다. 그 결과로, 이러한 궤적 데이터의 분석에 대한 관심이 증가하게 되었다 [3,4]. 모바일 컴퓨팅환경에서, 사용자는 단말기를 가지고 공간상에서 이동하며, 시간에 따른 그들의 위치를 무선 통신을 통해 시공간 데이터베이스에 저장하게 된다. 환경정보시스템에서의 동물들의 이동, 태풍 경로 추적과 같은 응용에서는 시간의 변화에 따른 객체들의 위치정보를 저장하기 때문에 대용량의 데이터를 만들어낸다. 컴퓨터 그래픽스 분야에서의 기본 연산은 유사한 패턴을 클러스터링하는 것이다 [6]. 시공간 데이터는 또한 세포 분열시 발생하는 미묘한 패턴의 발견에 집중하는 생물학에서 입자들의 이동에 따라 생겨날 수도 있다 [7].

이동 컴퓨팅 환경의 확산으로 (x, y, t) 로 표현할 수 있는 시공간상에서의 이동에 따라 궤적을 이루는 이동 객체를 효율적으로 처리할 수 있는 연구가 중요한 이슈가 되고 있다 [8]. 이동 객체의 궤적은 많은 정보를 포함하고 있기 때문에 이러한 궤적에 대한 분석은 많은 응용에서 관심있는 연구 주제중의 하나이다. 궤적 데이터의 분석을 위해 필요한 요구사항중의 하나는 유사한 궤적을 찾아내고 이를 클러스터링 하는 것이다. 예를 들어, “주어진 질의 궤적과 유사한 모든 이동 객체의 궤적을 찾아라”와 같은 질의가 가장 전형적인 예이다. 이동 객체를 위한 유사 검색은 도로 네트워크에서 이동 객체의 움직임 분석을 위해 필요한 중요한 기술이다. 따라서 이동 객체의 궤적 사이에서 유사도 측정 방법에 대한 정의가 필요하다. 유사도 측정은 유사궤적 검색과 유사한 궤적의 패턴으로부터 생겨날 수 있는 유용한 정보의 발견을 가능하게 한다. 예를 들어, 도로상의 자동차 궤적에 대한 분석을 바탕으로 마켓

팅 전략을 수립할 수 있다. 하지만, 기존의 대부분의 연구는 유클리디언 공간상에서의 유사 궤적 검색에 대한 연구이며, 도로 네트워크상의 이동 객체를 고려한 연구는 매우 적은 실정이다.

도로 네트워크의 특징을 고려하여, 현재 사용되고 있는 유사검색 방법들 [9-16]과 차별된 방법을 통해 유사 궤적을 검색하는 것이 필요하다. 기존의 방법들은 다음과 같은 단점을 가지고 있다. 첫째, 기존 연구들에서 가정한 유클리디언 거리 개념은 도로네트워크상에서는 정확하지 않다. 궤적의 유사도는 궤적간의 거리에 매우 의존적이기 때문에, 도로 네트워크에 적합한 유사도 측정 방법과 이를 기반으로 하는 유사궤적 검색 방법이 정의되어야 한다. 둘째, 대부분의 기존 연구들은 궤적의 시공간 특성이 아닌 공간적 특성만을 고려한 연구가 대부분이다. 예를 들어, 다른 시간에 같은 공간을 지나간 궤적 또한 유사 궤적으로 간주한다. 하지만 이는 시공간 특성상 유사한 궤적이 아니다. 아울러, 이동 객체의 궤적의 움직임이 정반대일지라도 유사한 궤적으로 간주한다.

도로 네트워크 상에서 이동 객체 궤적의 유사성 비교를 위해, 본 논문에서는 새로운 유사도 측정 방법을 제안한다. 이를 위해, 첫째, 두 궤적 사이의 공간 거리 계산을 위한 공간 유사도 측정 방법, 둘째, 시간 거리 계산을 위한 시간 유사도 측정 방법, 셋째, 이동 객체 궤적의 시공간 특성을 고려한 시공간 유사도 측정 방법을 제안한다. 제안하는 공간거리 계산은 기존 유클리디언 거리계산과는 달리 도로 네트워크상에서의 실제 네트워크 거리를 계산한다. 아울러, 시그니처 파일 기법을 이용하여 시공간 유사 궤적을 위한 새로운 유사 궤적 검색 알고리즘을 제안한다.

본 논문의 구성은 다음과 같다. 제 2장에서는 관련 연구를 소개한다. 제 3장에서는 제안하는 유사도 측정 방법들과 새로운 유사 궤적 검색 알고리즘을 설명하고, 제 4장에서는 실험을 통한 성능 분석 결과를 제시한다. 마지막으로 제 5장에서는 결론 및 향후 연구를 제시한다.

2. 관련 연구

유사 궤적 검색 문제는 수년 동안 많은 연구에서 관심 있는 주제였고, 이를 기반으로 많은 알고리즘이 개발되어 왔다. 기존 연구들은 주로 유사도를 측정하기 위한 거리 계산 방법 설계에 초점을 두었다. 본 논문의 관심사는 자동차와 같이 도로 네트워크 상에서 이동하는 이동 객체를 위한 유사 궤적을 검색하는 것이다. 본 절에서는, 유사 궤적 검색을 위한 관련 연구와 불필요한 데이터 검색을 효과적으로 줄일 수 있는 시그니처 파일 기법을 살펴본다.

[10]에서는 대용량의 이상값(outlier)를 포함하는 데이터에서 이동 객체의 시공간 궤적에 대한 분석 연구를 수행하였다. 유클리디언과 Time warping 방법의 성능을 개선하기 위해 LCSS(Longest Common Subsequence)에 기반한 비공간 거리 계산방법을 제안하였다. 하지만 유클리디언 거리에 기반한 연구이기 때문에, 도로네트워크 상에서의 유사 궤적 검색에는 적합하지 않다. [12]에서는, 시간과 공간을 모두 고려한 궤적으로부터 각 객체의 이동 패턴을 추출하기 위한 연구를 수행하였다. 이를 위해 패턴 분석을 위한 “상점에서 고객이 이동한 경로와 유사한 형태의 궤적을 갖는 모든 객체를 검색하여라”와 같은 궤적기반 질의를 제안하였다. 이러한 질의를 사용하여 상점에서 궤적의 형태에 따라 고객들을 분류하였다. 즉, 이들의 연구는 객체들이 이루는 궤적의 형태에 기반한 방법이다. 또한, 모바일 데이터의 궤적을 표현하기 위한 데이터 모델과 시계열 데이터베이스에서 시간 유사도를 확장하여 궤적 사이의 거리에 기반한 질의를 제시하였다. 아울러, 효과적인 검색을 위한 색인 기법을 제안하였다. 하지만 이 방법 역시 유클리디언 거리에 기반한 방법으로 도로 네트워크에 적용하기에는 문제가 있다. [13]의 연구는 시간을 고려하지 않은 이동 객체 궤적의 공간 형태에 초점을 맞추었다. 이들은 새로운 거리 개념인 OWD(one way distance)를 제안하고 이를 기반으로 단순하면서도 효과적으로 이동 객체의 궤적의 공간 형태를 비교하기 위한 방법을 제안하였다. 이를 이용하여 연속적인 궤적

은 piece wise linear 방법을 통해, 비연속적인 경우는 그리드 표현을 통해 궤적 사이의 OWD를 계산하여 효율적인 유사 궤적 검색 방법을 제안하였다. 그러나 궤적의 시간 특성을 고려하지 않았고, 유클리디언 공간상에서의 거리에 기반한 방법이기 때문에 실제 도로네트워크에서 궤적 사이의 거리를 계산하기에는 부적합하다. [14]의 연구에서는 분산 시공간 유사 검색 방법을 소개하였다. 주어진 궤적의 세그먼트가 다수의 노드에 분산 저장되었을 때, 사용자로부터 입력받은 k 개의 유사한 궤적을 검색하기 위해 LCSS에 기반한 UB-K (Upper Bound), UBLB-K (Upper Bound and Lower Bound)의 두 방법을 제안하였다. 제안된 방법은 시공간 특성을 모두 고려하였지만 유클리디언 공간을 가정하여 도로 네트워크 상에 적용할 수 없다는 단점이 있다. 이외에도 많은 유사 궤적 검색 연구들이 존재하지만 모두가 유클리디언 공간상에서 연구가 이루어져 도로 네트워크 상에 적용하기에는 부적합하다. [11,15,9,16].

도로 네트워크상에서 유사 궤적 검색을 위한 최초의 연구로 [17]이 존재한다. 시공간 특성을 고려하여 도로 네트워크 상에서 유사 궤적 검색을 위해 두 가지 요구 사항을 수립하고, 두 궤적 사이에 공통적으로 존재하는 POI(Points of Interest) 사이의 실제 네트워크 거리 및 시간 거리 기반으로 유사 궤적 검색 알고리즘을 제안하였다. 아울러 불필요한 궤적 검색을 효과적으로 줄이기 위해, 첫째, 질의 궤적의 POI를 모두 포함하지 않는 궤적을 필터링한 후 공통 POI 사이의 시간 거리의 합 계산, 둘째, 질의 궤적과 같은 시간 범위를 갖는 궤적을 검색한 후 TOI(Time of Interest) 사이의 공간 거리의 합 계산, 그리고 시공간기반 필터링 후 두 궤적의 속력을 이용한 시공간 거리의 합 계산의 3가지 방법을 제시하였다. 하지만, 질의 궤적과 공통된 POI를 모두 포함한 궤적만을 검색 대상으로 하기 때문에, 공간 거리상 매우 유사한 궤적일지라도 전체 POI를 포함하고 있지 않다면 검색대상에서 제외되어 정확한 유사 궤적 검색이 이루어지지 못하는 단점이 있다.

마지막으로 불필요한 데이터 검색을 효과적으로

줄일 수 있는 시그니처 파일 기법을 소개한다 [19,20]. 시그니처 파일 기법은 레코드를 구성하는 속성 값들의 비트 패턴을 계산하고 피트 패턴을 비트 OR 하여 레코드의 시그니처를 구성한 후 이를 시그니처 파일에 저장한다. 데이터를 검색할 시 먼저 질의 레코드에 대한 시그니처를 구성한 후 시그니처 파일에 저장된 각 레코드의 시그니처와 비트 AND 하여 질의 레코드 시그니처와 일치하는 후보 집합을 구한다. 따라서 질의 시그니처와 일치하지 않는 데이터들을 효과적으로 필터링할 수 있다. 마지막으로 후보 집합의 데이터를 검색하여 질의 데이터와 비교 후 질의 결과 집합을 구성하여 반환한다.

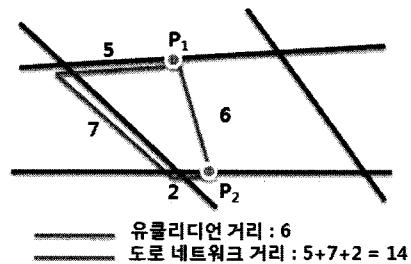
3. 도로 네트워크에서 이동 객체 궤적을 위한 유사도 측정 방법

본 논문에서는 기존 관련 연구들의 단점을 개선하기 위해 다음 두 가지 사항을 고려하고자 한다. 첫째, 대부분의 기존 연구 [9-16]는 공간상에서 객체 사이의 최단거리인 유클리디언 거리를 기반으로 한다. 그러나 이동 객체는 객체사이의 최단 네트워크 거리로 표시되는 도로 네트워크 상에서 움직이기 때문에 실제 네트워크 거리에 기반한 유사도 측정 방법이 필요하다. 둘째, 대부분의 기존 연구는 도로 네트워크 상에서 이동 객체 궤적의 공간 및 시간 특성 모두를 고려하지 않았다. [17]에서는 시공간 특성 모두를 고려하였지만 주어진 질의 궤적과 정확히 일치하거나 질의 궤적 전체를 포함하는 궤적만을 검색한다. 결과적으로, 질의 궤적과 부분적으로 일치하는 데이터 궤적은 유사 궤적 검색의 대상에서 제외된다.

따라서, 두 가지 고려 사항을 기반으로 본 절에서는 첫째, 도로 네트워크상에서 시공간 특성을 모두 고려한 유사도 측정 방법을 설계하며, 둘째, 질의 궤적과 부분적으로 일치하는 궤적 데이터 또한 검색할 수 있는 유사 궤적 검색 알고리즘을 제안하고자 한다.

3.1 공간 유사도 측정 방법

두 궤적의 공간 유사도는 두 궤적의 평균 공간 거리로 측정할 수 있다. 즉, 두 궤적의 평균 공간 거리가 가까울수록 보다 유사하다고 말할 수 있다. [18]에서는 이동 객체 궤적의 궤적을 구성하는 점들 사이의 평균 유클리디언 거리로 공간 유사도를 측정한다. 그러나 유클리디언 거리는 도로 네트워크에서의 두 점 사이의 실제 거리가 아니다. 그림 1은 유클리디언 거리와 도로 네트워크 거리의 차이를 나타낸다. 그림에 나타난바와 같이 P1에서 P2에 도착하기 위해서는 도로를 따라 이동해야하므로 실제 거리는 14이지만 두 점사이의 유클리디언 거리는 7로 정확한 거리를 나타내지 못한다. 따라서 도로 네트워크에서 궤적 사이의 공간 유사도는 두 궤적을 구성하는 각 점들의 최단 네트워크 거리인 공간 거리(SD)의 평균값으로 정의할 수 있다. 이때, 궤적은 여러 개의 에지로 구성되며, 에지는 양 끝에 노드(교차점)를 갖는 도로 세그먼트이다. 따라서, 두 궤적 사이의 공간 유사도를 측정하기 위해, 첫째, 질의 궤적과 데이터 궤적 모두 에지 단위로 나눈다. 이때, 궤적의 각 에지에 대하여, 데이터 궤적의 에지 중 공간적으로 가장 가까운 에지가 해당 질의 궤적 에지의 쌍을 이룬다. 이를 통해 질의 궤적과 데이터 궤적 사이의 에지 쌍의 시퀀스를 구한다. 질의 궤적과 데이터 궤적 사이의 공간 유사도는 구해진 에지 쌍의 평균 공간 거리이다. 이를 위해 도로 네트워크 상에서 각 도로의 두 노드를 이용하여 에지 사이의 공간 거리를 정의하고, 공간 유사 검색을 위한 새로운 공간 유사도 측정 방법을 정의한다.



<그림 1> 유클리디언 거리와 도로 네트워크의 거리의 차이

3.1.1 공간 유사도(SS)

본 절에서는 공간 거리(SD)에 기반한 공간 유사도(SS)를 정의한다. 질의 궤적(Q)과 데이터 궤적(R) 이 $Q = \{q_1, q_2, q_3, \dots, q_n\}$, $R = \{r_1, r_2, r_3, \dots, r_m\}$ 과 같다고 가정하자. 이때, 두 궤적을 구성하는 에지 사이의 공간 거리 $D_{edge}(Q_e, R_e)$ 는 다음과 같이 각 에지의 시작 노드 사이의 네트워크 최단 거리(D_n)와 끝 노드 사이의 네트워크 최단 거리의 합을 노드 개수인 2로 나눈 평균 거리로 정의하며, D_n 은 네트워크상의 실제 거리를 말한다.

$$D_{edge} = \frac{D_n(SNode\ of\ q_e, SNode\ of\ r_e) + D_n(ENode\ of\ q_e, ENode\ of\ r_e)}{2}$$

정의 1: 두 궤적 Q와 R이 주어졌을때, Q와 R의 매칭 에지 쌍 $MEdgePair(Q,R)$ 은 궤적 Q의 에지 q_e 와 가장 작은 에지 공간 거리 $Dedge(q_e, r_e)$ 를 갖는 궤적 R의 에지 r_e 의 쌍의 집합으로 정의한다. 여기에서 e 는 궤적 Q와 R사이의 매칭 에지 쌍의 개수이다.

$$MEdgePair(Q,R) = \{(q_e, r_e, Dedge(q_e, r_e)), 1 \leq e \leq m\}$$

정의 2: 두 궤적 Q와 R의 공간 유사도(SS)는 매칭 에지 쌍의 평균 도로 네트워크 거리로 정의한다. 이때, m 은 궤적 Q와 R사이의 매칭 에지 쌍의 개수이다.

$$SS(Q, R) = \frac{1}{m} \left[\sum_{k=1}^m D_{edge}(q_k, r_k) \right]$$

3.1.2 공간 유사도 측정 알고리즘

그림 2는 도로 네트워크에서 두 궤적 사이의 공간 유사도 측정 알고리즘을 나타낸다. 첫째, 두 궤적을 위한 매칭 에지 쌍 집합과 공간 거리의 초기화를 수행한 후 궤적 사이의 매칭 에지 쌍을 구한다. 둘째, 서로 다른 길이의 궤적 사이의 유사도 측

정을 지원하기 위해 먼저 주어진 질의 궤적의 각 에지와 가장 가까운 데이터 궤적 R의 에지를 검색하여 쌍을 구한 후, 질의 궤적의 에지와 쌍을 이루지 않는 데이터 궤적 R의 나머지 에지와 가장 가까운 질의 궤적 에지를 검색하여 쌍을 구한다. 셋째, 검색된 모든 매칭 에지 쌍 사이의 평균 거리 계산을 수행한다. $findMatchEdge(q_e, R, Medge)$ 함수는 주어진 에지 q_e 와 궤적 R의 에지중 에지 공간 거리가 가장 작은 에지를 검색하여 에지 공간 거리와 함께 매칭 에지 쌍 집합에 삽입한다. 이때, 에지 사이의 공간 거리는 에지를 구성하는 각 시작 노드 사이의 최단 거리와 끝 노드 사이의 최단 거리의 평균값으로 구한다. 아울러 최단 거리 계산은 매우 많은 비용을 요구하므로 Dijkstra 알고리즘을 이용하여 도로 네트워크를 구성하는 각 노드 사이의 최단 거리를 미리 계산하여 저장한다. 넷째, 두 거리 저장 변수인 $minDistS$ (가능한 에지 쌍중 에지간 최소 시작 노드 거리)와 $minDistE$ (가능한 에지 쌍중 에지간 최소 끝 노드 거리)의 초기화를 수행한다. 다섯째, 에지 q_e 와 동일한 시작 노드, 끝 노드를 가진 매칭 에지를 검색하거나 모든 데이터 궤적 에지와 계산을 마칠때까지 반복된다. 이때, 동일한 시작, 끝 노드를 가지지만 지하도로나 도로 내 고가 다리 또는 우회 도로와 같은 다른 에지가 존재 할 수 있다. 이러한 경우의 공간 에지 거리는 두 에지간 평균 도로 네트워크 거리인 $1/3(q_e.length + r_e.length)/2$ 로 정의 한다. 마지막으로, 만약 동일한 시작 노드 및 끝 노드를 갖는 데이터 궤적의 에지가 존재 하지 않을 경우, 가장 가까운 시작 노드 거리를 가진 에지와 가장 가까운 끝 노드 거리를 가진 에지 중 보다 적은 공간 에지 거리를 갖는 에지를 매칭 에지 쌍으로 선택한다.

그림 3은 질의 궤적 Q와 데이터 궤적 R, S사이의 공간 유사도 측정 예를 나타낸다. 이때 질의 궤적의 방향은 왼쪽에서 오른쪽으로 진행된다고 가정한다. 그림에서 질의 궤적은 $Q = \{q_1, q_2, q_3, q_4, q_5, q_6\}$ 의 6개 에지로 구성되고, 데이터 궤적은 각각 $R = \{r_1, r_2, r_3, r_4, r_5, r_6, r_7\}$, $S = \{s_1, s_2, s_3, s_4, s_5, s_6\}$ 로 7, 6개의 에지로 구성

```

SpatialSimilarity(Q,R) {
Input: Q(질의 쿼지), R(데이터쿼지)
Output: 두 쿼지 Q와 R사이의 공간 거리(SD)
    Medge = { ∅ }; // 매칭 에지 쌍 집합 (qe, re, Dedge)
    SD = 0; // 공간 거리
    for every edge of Q, qe
        findMatchEdge(qe,R,Medge); //qe: 쿼지 Q의 에지(e=1,2,3,.....n)
    for unselected edges of R
        findMatchEdge(re,Q,Medge); //re: 쿼지 R의 에지(e= 1,2,3,.....m)
    for all n number of matching edge pairs(Q,R)
        SD = Dedge(qe, re);
        SD = SD/n;
}

findMatchEdge(qe, R, Medge) {
    minDistS=minDistE= ∞ ;
    for every edge of R, re {
        if (Dn(qe.StartN, re.StartN) < minDistS)
            { minDistS= Dn(qe.StartN, re.StartN); minStartN= re ;}
        if (Dn(qe.EndN, re.EndN) < minDistE)
            { minDistE= Dn(qe.EndN, re.EndN); minEndN= re ;}
        if (minDistS== 0 && minDistE == 0) && qeID== reID
            { Medge = Medge U < qe, re, 0 >; return;}
        else
            { Medge = Medge U < qe, re, 1/3(qe.length + re.length)/2>; return;}
    }
    if ((Dn(Q.StartN, R.StartN) - Dn(Q.StartN,R.EndN) > 0 )
        Medge= Medge U < qe, re, Dn(qe.StartN, re.EndN) >;
    else
        Medge= Medge U < qe, re, Dn(qe.StartN, re.StartN) >;
}

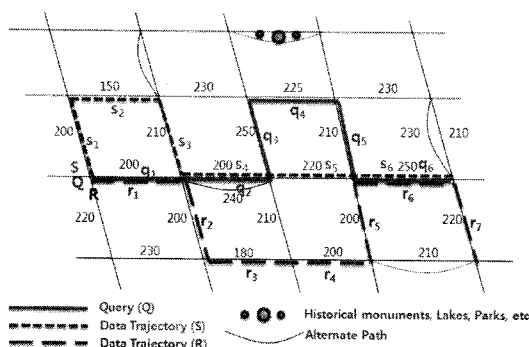
```

〈그림 2〉 공간 유사도 측정 알고리즘

되며 숫자는 네트워크 거리를 나타낸다. 먼저 질의 쿼지 Q와 데이터 쿼지 R 사이의 매칭 에지 쌍 집합을 구한다. 이를 위해 질의 쿼지의 첫 번째 에지인 q1과 데이터 쿼지 R의 각 에지와의 공간 에지 거리를 구한다. 이때 q1과 r1은 동일한 에지로서 공간 에지 거리는 0이므로 q1의 매칭 에지 쌍은 r1이 된다{(q1, r1, 0)}. 질의 쿼지의 두 번째 에지인 q2에서는 r2가 가장 가까운 시작 노드 거리를 갖는 에지로서 q2의 매칭 에지 쌍으로 선택된다. 이때 두 에지 사이의 공간 에지 거리는 $(0+390)/2=195$ 가 된다. 따라서 매칭 에지 쌍 집합은 {(q1,r1,0),(q2,r2,195)}이 된다. 같은 방법으로 질의 쿼지의 나머지 에지 q3~q6에 대한 매칭 에지 쌍 집합을 구하면, $MedgePair(Q,R) = \{(q1, r1,$

0), (q2, r2, 195), (q3, r3, 425), (q4, r4, 435), (q5, r5, 205), (q6, r6, 0)}이 된다. 마찬가지로 데이터 쿼지 S에 대한 매칭 에지 쌍 집합 $MedgePair(Q,S) = \{(q1, s3, 175), (q2, s4, 0), (q3, s5, 218), (q4, s5, 230), (q5, s5, 215), (q6, s6, 0)\}$ 이 된다. 이때, 데이터 쿼지 S의 에지 s1, s2와 R의 에지 r7은 질의와의 매칭 에지 쌍이 없기 때문에, 이에 대한 질의 쿼지 Q의 매칭 에지 쌍을 계산하면 각각 q1, q1, q6이 매칭 에지 쌍이 된다. 따라서 매칭 에지 쌍 집합은 $MedgePair(Q,R) = \{(q1, r1, 0), (q2, r2, 195), (q3, r3, 425), (q4, r4, 435), (q5, r5, 205), (q6, r6, 0), (r7, q6, 235)\}$, $MedgePair(Q,S) = \{(q1, s3, 175), (q2, s4, 0), (q3, s5, 218), (q4, s5, 230),$

(q5, s5, 215), (q6, s6, 0), (s1, q1, 180), (s2, q1, 205)} 이 된다. 마지막으로, 검색된 매칭 에지 쌍 집합의 평균 공간 에지 거리를 계산한다. 따라서, 질의 궤적 Q와 데이터 궤적 R의 공간 유사도 $SS = (0+195+425+435+205+0+235)/7 = 213$ 이 되며, Q와 S의 공간 유사도 $SS' = (175+0+218+230+215+0+180+205)/8 = 152$ 가 되어, 질의 궤적 Q에 공간적으로 유사한 궤적은 공간 유사도 값이 적은 궤적 S가 된다.



<그림 3> 질의 궤적 Q와 데이터 궤적 R, S의 예

3.2 시간 유사도 측정 방법

시간은 정보의 신뢰성, 유효성, 적용 가능성 면에서 중요한 역할을 수행하기 때문에 도로 네트워크에서 유사 궤적 검색을 위해 공간 유사도와 마찬가지로, 궤적의 시간 유사도 정의가 필요하다. 두 궤적의 시간 유사도는 두 궤적의 각각의 시간 차이를 시간 거리로 계산하여 측정한다. 따라서 질의 궤적과 보다 적은 시간 거리를 가진 데이터 궤적이 시간적으로 보다 유사한 궤적이다. 시간 거리를 계산하기 위해 시간을 하루의 시간 범위(R), 요일(D), 주(W)의 3가지 범위로 나눈다. 먼저, R, D, W의 시간 거리를 정의하고 이들의 합을 궤적의 시간 거리로 정의한다.

3.2.1 영역 거리에 기반한 시간 유사도

하루 24시간중 도로 네트워크에는 많은 교통 상

황이 발생한다. 첫째, 아침과 저녁 러시아워에 출퇴근으로 인해 교통 체증이 발생한다. 둘째, 낮과 늦은 저녁에는 이동 객체의 수가 줄어 교통 체증 또한 줄어든다. 셋째, 자정 무렵에는 이동 객체의 감소로 교통 체증 또한 거의 존재하지 않는다. 이를 통해, 같은 도로에서도 하루 24시간의 변화에 따라 도로 환경 또한 다르다는 결론을 내릴 수 있다. 도로 네트워크에서의 이러한 상황을 반영하기 때문에 시간에 따른 이동 객체의 궤적 패턴이 다양하게 나타난다. 따라서, 하루를 표 1과 같이 7개의 시간 영역으로 분할 할 수 있다. 아울러, 영역에 따른 비슷한 교통 상황에 따라 표 2에 나타난 바와 같이 4개의 그룹으로 분할할 수 있다.

나누어진 시간 영역을 기반으로 영역 사이의 거리가 0-7의 값을 갖는 영역 거리(RD)를 정의하고 이를 바탕으로 궤적의 시작 시간 영역에서 종료 시간 영역까지의 영역 거리로부터 시간 영역 거리(TRD)를 정의한다. 도로 네트워크에서 질의 궤적과 데이터 궤적을 각각 Q, T라 가정할 때, 궤적 Q는 시간 영역 Qst에서 시작하여 Qet에서 끝나고, 궤적 T는 시간 영역 Tst에서 시작하여 Tet에서 끝난다.

<표 1> 7개의 시간 영역

번호	시간 영역 (R)	설 명	시 작	끝
1.	MR	아침 러시아워	7:00 AM	9:00 AM
2.	M	아 침	9:00 AM	12:00 PM
3.	L	점 심	12:00 PM	2:00 PM
4.	A	오 후	2:00 PM	5:00 PM
5.	ER	저녁 러시아워	5:00 PM	7:00 PM
6.	LE	늦은 저녁	7:00 PM	10:00 PM
7.	N	밤	10:00 PM	7:00 AM

<표 2> 시간 영역의 4 그룹

그 룹	시간 영역 구성 멤버
그룹 1	MR, ER
그룹 2	L
그룹 3	M, A, LE
그룹 4	N

정의 3: 시간 영역 거리

궤적 Q, T가 동일한 시간 영역에서 이동했을 때, 두 궤적 사이의 영역 거리 RD(Q,T)는 0으로 정의하며, 다른 시간 영역에서 이동했지만, 같은 그룹에 속하였을 때는 1로 정의한다. 만약 궤적 Q, T가 다른 시간 영역 그룹에 속해 있을 때, 영역 거리 RD(Q,T)는 다음과 같다.

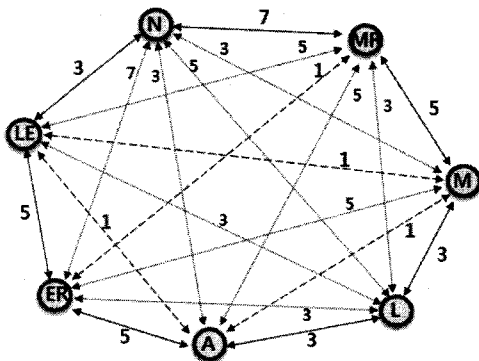
$$RD(Q,T) = |그룹_i - 그룹_j| * 2 + 1, \\ 1 \leq i \leq 4, 1 \leq j \leq 4$$

정의 4: 궤적 시간 영역 거리

궤적 Q, T가 각각 한 시간 영역에서 시작하여 다른 시간 영역에서 끝났을 때 두 궤적 사이의 시간 영역 거리 TRD(Q,T)는 시작 영역 거리와 종료 영역 거리의 합을 2로 나눈 평균 영역 거리로 정의한다.

$$TRD(Q,T) = (RD(Q_{st}, T_{st}) + RD(Q_{et}, T_{et})) / 2,$$

그림 4는 모든 가능한 쌍의 영역 거리를 나타낸다. 그림에서 실선은 연속된 영역 사이의 영역 거리를 나타내며, 가는 점선은 같은 그룹내의 영역 거리를, 마지막으로 굵은 점선은 다른 그룹에 존재하는 영역 사이의 거리를 나타낸다.



<그림 4> 시간 영역에 따른 영역 거리

3.2.2 날짜 거리에 기반한 시간 유사도 (D)

일주일은 월요일부터 일요일까지의 7개의 요일로 구성된다. 7개의 요일 중 월요일에서 금요일까지는 근로일, 토요일과 일요일은 주말로 구분한다. 이동 객체의 궤적은 근로일과 주말에 다른 궤적 패턴을 나타낸다. 따라서 일주일을 아래와 같이 두개의 그룹으로 나눌 수 있다.

Group1 = 근로일 = 월, 화, 수, 목, 금

Group2 = 주말 = 토, 일

이를 바탕으로 두 궤적의 날짜 거리(DD)를 일주일의 각 요일 사이의 거리인 Day-dist와 근로일, 주말 그룹 사이의 거리인 Group-dist의 합으로 정의한다. 궤적 Q가 Qsd 일에 시작하여 같은 날인 Qed에 종료하고, 궤적 T가 Tsd일에 시작하여 같은 날인 Ted에 종료한다고 가정한다.

정의 5: 요일 거리

만약 궤적 Q, T가 같은 날에 존재한다면 요일 거리 Day-dist(Q,T)는 0으로 정의하며, 다른 날에 존재한다면 요일들의 원형 순서 상에서의 요일 차이로 나타낸다. 따라서 날짜 거리 Day-dist(Q,T)는 1-3의 값을 갖는다.

정의 6: 그룹 거리

만약 궤적 Q, T가 같은 요일 그룹에 존재한다면 궤적 사이의 그룹 거리 Group-dist(Q,T)는 0이며, 그렇지 않으면 그룹 거리 Group-dist(Q,T)는 5이다.

정의 7: 날짜 거리

만약 Q, T가 다른 그룹에 존재한다면 두 궤적 사이의 날짜 거리 DD(Q,T)는 다음과 같이 정의 한다.

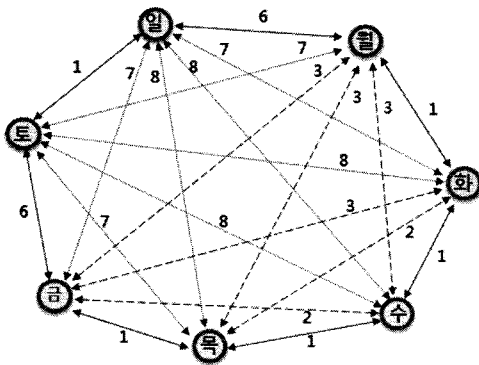
$$DD(Q, T) = Group-dist(Q,T) + Day-dist(Q,T)$$

정의 8: 궤적 날짜 거리

궤적 Q, T의 시작과 종료가 각각 다른 요일에 존재한다면 두 궤적 사이의 날짜 거리 TDD(Q,T)는 시작일의 날짜 거리와 종료일의 날짜 거리의 합을 2로 나눈 평균 날짜 거리로 정의한다.

$$TDD(Q, T) = (DD(Q_{sd}, T_{sd}) + DD(Q_{ed}, T_{ed})) / 2$$

그림 5는 나타낼 수 있는 모든 가능한 날짜 거리를 나타낸다. 그림에서 실선은 두 요일 사이의 날짜 거리를 나타내며, 가는 점선은 다른 그룹에 속하는 요일 사이의 날짜 거리를 굵은 점선은 같은 그룹에 속하는 요일 사이의 날짜 거리를 나타낸다.



〈그림 5〉 7개 요일 사이의 날짜 거리

3.2.3 주에 기반한 시간 유사도 (W)

시간 범위를 날짜 거리에서 년도와 같은 보다 넓은 범위로 확장하기 위하여 시간 유사도의 범위를 확장한다. 이를 위해 요일보다 넓은 범위의 시간 개념인 월, 년도는 주(week)로 표시할 수 있기 때문에 주를 이용한 주간 거리(WD)를 정의한다. 예를 들어, 각각 한달은 4주, 1년은 52주 단위로 나타낼 수 있다. 이를 통해 월이나 년도를 구분하기 위한 비용을 줄일 수 있다. 궤적 Q가 Q_{sw}에서 시작하여 Q_{ew}에서 종료하고, 궤적 T가 T_{sw}에서 시작하여

T_{ew}에서 종료한다고 가정한다.

정의 9: 주간 거리 WD(Q, T)는 두 주사이 차이의 절대값으로 정의한다.

정의 10: 궤적 Q, T가 존재할 때, 두 궤적 사이의 궤적 주간 거리 TWD(Q,T)는 두 궤적의 시작 주와 종료 주의 차이의 합을 2로 나눈 평균 주간 거리로 정의한다.

$$TWD(Q,T) = (WD(Q_{sw}, T_{sw}) + WD(Q_{ew}, T_{ew})) / 2$$

3.2.4 영역 거리, 날짜 거리, 주간 거리를 통합한 시간 유사도

정의 11: 도로 네트워크 상에서 두 궤적 Q, T 사이의 시간 거리 (TD)는 범위 거리 TRD, 날짜 거리 TDD, 주간 거리 TWD의 합으로 정의한다. 이때, α, β, γ 는 시간 가중치이다.

$$TD(Q, T) = \alpha * TRD + \beta * TDD + \gamma * TWD \quad (\text{식 1})$$

두 궤적 사이의 시간 거리에 각각 α, β, γ를 사용한 이유는 다음과 같다. 주어진 궤적 Q, T1이 같은 시간 영역, 같은 요일이지만 다른 주에 존재하면 같은 범위, 같은 주이지만 다른 요일에 존재하는 궤적 T2보다 유사하다. 다시 말하면 시간 유사도에서 날짜 거리가 주간 거리보다 높은 우선순위를 갖는다. 또한, 주어진 궤적 Q, T1이 다른 시간 영역에 존재하면 요일이나 주간 거리에 상관없이 같은 시간 영역에 존재하는 궤적 T2보다 덜 유사하다. 다시 말하면, 영역 거리가 시간 유사도에서 날짜 거리와 주간 거리보다 높은 우선순위를 갖는다. 따라서, 궤적의 시간 유사도를 영역 거리, 날짜 거리, 주간 거리의 순으로 계산하기 위하여 영역 거리 RD에 가장 높은 우선순위를, 날짜 거리에 두 번째 우선순위를, 주간 거리에 가장 낮은 우선순위를 준다.

그러나, 앞선 시간 유사도 설계에서 영역 거리는 0-7, 날짜 거리는 0-8, 주간 거리는 0-259의 값을 각각 갖기 때문에 각 시간거리에 이러한 우선순위를 반영하기 위하여 3개의 시간 가중치인 α , β , γ 를 사용하며, 그 값은 다수의 실험을 통하여 $\alpha = 60$, $\beta = 20$, $\gamma = 1$ 으로 정의한다. 따라서, 시간 거리 계산은 다음과 같다.

$$TD(Q, T) = 60 * TRD + 20 * TDD + TWD \quad (\text{식 } 2)$$

식 (2)에 따라, 시간 거리 TD는 $60 * 7 + 20 * 8 + 259 = 420 + 160 + 259 = 839$ 의 최대값을 갖는다. 따라서 시간 거리의 범위는 0-839이다.

3.2.5 시간 유사도 측정 알고리즘

그림 6은 시간 유사도 측정 알고리즘을 나타낸다. 시간 유사도 측정 알고리즘은 질의 궤적과 데이터 궤적 사이의 시간 거리를 계산하여 반환한다. 첫째, getTemporalDistance 함수를 이용하여 질의 궤적 Q와 데이터 궤적 Dt사이의 영역거리 TRD, 날짜 거리 TDD, 주간 거리 TWD를 계산하여 반환한다. 둘째, 계산된 TRD, TDD, TWD를 각각의

가중치인 $\alpha (= 60)$, $\beta (= 20)$, $\gamma (= 1)$ 를 이용하여 궤적 Q, T사이의 시간 유사도인 시간 거리를 계산하여 반환한다.

3.3 시공간 유사도 측정 방법

3.1 및 3.2절에서 두 궤적 사이의 공간 유사도 측정과 시간 유사도 측정을 위해 공간 거리 SD, 시간 거리 TD를 각각 정의하였다. 본 절에서는, 두 가지 측정 방법을 이용하여 이동 객체 궤적을 위한 시공간 유사도 측정 방법을 정의한다. 이를 위해 두 궤적 사이의 시공간 거리 STDist를 계산하기 위한 알고리즘을 설계한다.

3.3.1 시공간 유사도

공간 유사도와 시간 유사도를 이용하여 시공간 유사도 측정 방법을 설계하기 위해 두 가지 접근방법을 사용한다. 첫째, 합 접근 방법과, 둘째, 곱 접근 방법이다. 두 가지 접근 방법은 적용하고자 하는 응용에 따라 선택하여 사용할 수 있다. 다음 정의를 위해 도로 네트워크 상의 질의 궤적을 Q, 데이터 궤적을 T라 가정한다.

```

TemporalSimilarity(Q, Dt) {
    Input: 질의 궤적 Q, 데이터 궤적 Dt
    Output: 궤적 Q, Dt사이의 시간 거리(TD)
        temporalDistance=0;
        getTemporalDistance(Q, Dt, TRD, TDD, TWD);
        temporalDistance =  $\alpha$ *TRD +  $\beta$ *TDD +  $\gamma$ * TWD
        return temporalDistance;
}

getTemporalDistance(Q, Dt, TRD, TDD, TWD) {
    QR , QD , QW //range, day and week distances of Q
    DtR, DtD, DtW; //range, day and week distances of Dt
    TRD = 0, TDD = 0, TWD = 0;
    TRD = RD(QR, DtR);
    TDD = DD(QD, DtD)
    TWD = WD(QW, DtW)
}
    
```

<그림 6> 시간 유사도 측정 알고리즘

정의 12: 합 접근 방법을 이용한 시공간 거리 질의 궤적 Q와 데이터 궤적 T 사이의 시공간 거리 STD_{Dist} 는 두 궤적 사이의 공간 거리 SD와 시간 거리 TD의 평균값으로 정의한다. 여기서 δ 는 시간 거리를 위한 가중치이다.

$$STD_{Dist}(Q,T) = (SD + \delta * TD) / 2 \quad (\text{식 3})$$

가중치 δ 는 다음과 같이 정의할 수 있다. 앞서 정의한 시간 거리 TD의 최대값은 839임을 계산해 통해 알 수 있다. 그러나 공간 거리 SD의 경우, 도로 네트워크내의 세그먼트 수에 비례하여 시간 거리보다 훨씬 높은 값을 갖는다. 즉, 도로 네트워크를 위한 지도의 크기가 커질수록 공간 거리의 최대값 또한 비례하여 증가한다. 따라서, 시공간 거리 계산에서 공간 거리와 시간 거리가 각각 같은 비중을 차지할 수 있도록 보다 적은 값을 갖는 시간 거리에 두 거리의 최대값 비율에 따른 가중치를 곱하여 정확한 시공간 거리를 계산한다. 가중치 δ 는 다음과 같이 정의한다.

$$\delta = \text{공간 거리 최대값} / \text{시간 거리 최대값}$$

식 (3)을 사용하여 계산된 합 접근 방법을 이용하여 계산된 시공간 거리는 "러시아워 동안에 최적의 경로를 탐색하여라"와 같은 시간 도메인이 공간 도메인 보다 중요한 응용에 사용될 수 있다.

정의 13: 곱 접근 방법을 이용한 시공간 거리 질의 궤적 Q와 데이터 궤적 T 사이의 시공간 거리 STD_{Dist} 는 두 궤적 사이의 공간 거리 SD와 시간 거리 TD의 곱로 정의한다. 여기서 δ 는 공간 거리를 위한 가중치이다.

$$STD_{Dist}(Q,T) = ((SD / \delta) + 1) * (TD + 1) \quad (\text{식 4})$$

곱 접근 방법에서의 공간 거리를 위한 가중치 δ

는 합 접근 방법과 같은 방법으로 계산할 수 있다. 공간 거리 및 시간 거리에 1을 합하는 이유는 시간 거리나 공간 거리가 0이 되었을 때, 전체 시공간 거리가 0이 되어 부정확한 거리값이 생성되는 것을 막기 위함이다. 1은 공간 거리나 시간 거리의 최대값에 비해 매우 작은 값으로 전체 시공간 거리에 영향을 거의 미치지 않는다. 식 (4)를 사용하여 계산된 곱 접근 방법을 이용한 시공간 거리는 "새로운 비즈니스를 시작하기 위한 최적의 장소를 찾기 위한 비즈니스 전략"과 같은 공간 도메인이 시간 도메인보다 중요한 응용에 사용될 수 있다.

3.3.2 시공간 유사도 측정 알고리즘

그림 7은 주어진 질의 궤적 Q와 데이터 궤적 DT 사이의 시공간 유사도를 측정하는 알고리즘이다. 이를 위해 spatio-temporal distance(STD_{Dist}) 함수는 시공간 유사도의 척도인 시공간 거리를 계산하여 반환한다. 시공간 유사도 측정 알고리즘은 첫째, 공간 유사도 측정 알고리즘을 이용하여 주어진 질의 궤적 Q와 데이터 궤적 T사이의 공간 거리를 계산한다. 둘째, 시간 유사도 측정 알고리즘을 이용하여 시간 거리를 계산한다. 마지막으로, 계산된 공간 거리와 시간 거리를 이용하여, 곱 접근 방법이나 합 접근 방법에 따라 시공간 거리 STD_{Dist} 를 계산한다.

3.4 시공간 유사 궤적 검색 알고리즘

본 절에서는 시공간 유사 궤적 검색 알고리즘을 설계한다. 제안하는 시공간 유사 궤적 검색 알고리즘은 효과적인 검색을 위하여 시그니처 기반 액세스 기법을 사용한다. 이를 통해 불필요한 데이터 궤적의 검색을 효율적으로 줄일 수 있다.

3.4.1 시그니처 기반 궤적 탐색 알고리즘

도로 네트워크의 교통 분석이나 과거 궤적 데이터를 기반으로 최적의 경로를 검색하기 위한 응용을 지원하기 위해서는 주어진 질의 궤적과 부분적

```

Spatio-TemporalSimilarity(Q, DT) {
Input: 질의 궤적 Q, 데이터 궤적 DT
Output: 궤적 Q, DT사이의 시공간 거리 STDist
    SD=SpatialSimilarity(Q, DT) // 공간 유사도 측정 알고리즘 이용
    TD=TemporalSimilarity(Q, DT) // 시간 유사도 측정 알고리즘 이용
    Compute Spatio-temporal Distance (STDist) using one of below two
    (A) 곱 접근 방법
        STDist(Q, Dt) = ( (SD/δ)+1)*(TD+1)
    OR
    (B) 합 접근 방법
        STDist(Q, Dt) = (SD+ δ*TD)/2
}
    
```

<그림 7> 시공간 유사도 측정 알고리즘

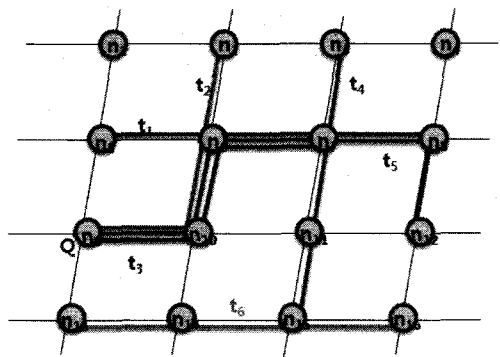
```

Signature_based Trajectory Search Algorithm(Q, DT) {
Input: 질의 궤적 Q={S1, S2, ...Sn}, 데이터 궤적 집합 DT={T1, T2, ...Tm}
Output: 궤적 Q와 한 개 이상의 일치하는 세그먼트를 갖는 데이터 궤적 집합
    Candidate = ∅ ;
    For each Si in Q
        Qsignature[i]=generate_signature(Si)
    for each Ti in DT
        for each Sj in Q {
            If ( signature_match(Sj, Ti);
                { Candidate = Candidate U Ti;      Break; }
        }
    Return Candidate;
}
    
```

<그림 8> 시그니처 기반 궤적 탐색 알고리즘

으로 일치하는 유사 궤적을 검색하여 이를 데이터 마이닝 기법을 이용하여 시간별, 요일별 도로의 평균 운행 패턴에 대한 분석이 필요하다. 이를 위해 주어진 질의 궤적에 대한 후보 집합을 검색하기 위하여 질의 궤적과 적어도 하나 이상의 일치하는 세그먼트를 갖는 유사 데이터 궤적을 탐색하여야 한다. 이를 효과적으로 탐색하기 위해 시그니처 기반 궤적 탐색 알고리즘을 설계한다. 그림 8은 시그니처 기반 궤적 탐색 알고리즘을 나타낸다. 첫째, 주어진 궤적 Q의 각 세그먼트에 대한 시그니처를 생성한다. 둘째, 데이터 궤적 시그니처와 질의 궤적의 세그먼트 시그니처를 비교하여 일치하면 후보 집합에 삽입하고, 그렇지 않으면 질의 궤적의 다음 세그

먼트 시그니처와 비교한다. 마지막으로 탐색된 후보 집합을 반환한다.



<그림 9> 질의 궤적 Q와 데이터 궤적의 예

예를 들어, 그림 9와 같은 질의 궤적 Q와 데이터 궤적들이 주어졌을 때를 가정하자. 그림에서 각 궤적의 이동 방향은 좌측에서 우측방향 또는 상단에서 하단 방향이다.

모든 궤적들은 표 3에 나타난바와 같은 도로 세그먼트 순서로 구성된다. 예를 들어 질의 궤적 $Q = \{s17, s13, s9\}$ 의 3개의 세그먼트로 구성된다. 궤적 탐색을 위해 첫째, 질의 궤적의 각 세그먼트에 대한 시그니처를 생성한다. 질의 궤적에 대한 시그니처는 표 4에 나타난바와 같이 $Q_{signature} = \{00100010, 01000001, 01010000\}$ 이 된다. 둘째, 데이터 궤적 $t1-t6$ 에 대하여 $Q_{signature}$ 의 시그니처와 일치하는 궤적을 갖는 데이터 궤적을 탐색한다. 질의 궤적의 첫 번째 세그먼트의 경우 탐색되는 궤적은 $t2, t3$ 이며 나머지 세그먼트와 일치하는 데이터 궤적은 표 5와 같다. 따라서 궤적 탐색을 통해 생성되는 후보 집합 $Candidate = \{t1, t2, t3\}$ 이 된다.

〈표 3〉 질의 궤적과 데이터 궤적의 세그먼트 구성

Trajectories	Segments
Q_t	{s17, s13, s9}
$t1$	{s10, s9, s8}
$t2$	{s17, s13, s6}
$t3$	{s17, s13, s9}
$t4$	{s5, s12, s19}
$t5$	{s8, s11}
$t6$	{s22, s23, s24}

〈표 4〉 각 세그먼트에 대한 시그니처

Road Network Segments	Segment Signature	Road Network Segments	Segment Signature
s1	11000000	s13	01000001
s2	10100000	s14	00110000
s3	10010000	s15	00101000
s4	10001000	s16	00100100
s5	10000100	s17	00100010
s6	10000010	s18	00100001
s7	10000001	s19	00011000
s8	01100000	s20	00010100
s9	01010000	s21	00010010
s10	01001000	s22	00010001
s11	01000100	s23	00001100
s12	01000010	s24	00001010

〈표 5〉 시그니처 파일을 이용한 부분 검색 결과

Query Segment Signature	Matching Data Trajectories
s17 = 00100010	<t2, t3>
s13 = 01000001	<t2, t3>
s9 = 01010000	<t1, t2>

3.4.2 시공간 유사 궤적 검색 알고리즘

주어진 질의 궤적과 시공간상 가장 유사한 궤적을 검색하기 위해서 효율적으로 후보 집합을 검색할 수 있는 방법이 필요하다. 이를 위해 시그니처 파일 기법을 사용한다[19,20]. 그림 10은 시공간 유사 궤적 검색 알고리즘을 나타낸다.

그림 10에서, 첫째, 주어진 질의 궤적 Q의 세그먼트에 대하여 한 개 이상의 동일한 세그먼트를 갖는 궤적을 탐색한다. 미리 계산된 궤적들의 시그니처에 대하여 질의 궤적을 이루는 각 세그먼트의 시그니처를 이용하여 탐색한다. 이때, 질의 궤적 세그먼트와 일치하는 궤적을 후보 집합에 삽입한다. 둘째, 후보 집합의 모든 궤적에 대하여 공간 거리 SD와 시간 거리 TD를 각각 계산한다. 셋째, 계산된 공간 거리와 시간 거리를 이용하여 접근 방법에 따라 시공간 거리 $SDist$ 를 계산한다. 넷째, 계산된 시공간 거리 $SDist$ 를 이용하여 후보 집합을 정렬한다. 마지막으로 정렬된 후보 집합에서 질의 궤적과 가장 유사한 k 개의 결과를 반환한다.

예를 들어 앞 절의 그림 9와 같이 질의 궤적 Q와 6개의 데이터 궤적이 주어졌을 때 표 6은 시공간 유사 궤적 검색 알고리즘에 의한 유사 궤적 검색의 예를 나타낸다. 주어진 질의 궤적 Q에 대하여, 첫째, 시그니처 기반 액세스 기법을 이용하여 6개의 데이터 궤적 중 질의 궤적의 세그먼트와 한 개 이상의 세그먼트와 일치하는 데이터 궤적 $t1, t2, t3$ 를 탐색하여 후보 집합을 생성한다. 둘째, 후보 집합 $Result1 = \{t1, t2, t3\}$ 에 대하여 질의 궤적과의 공간 거리 SD, 시간 거리 TD를 계산한다. 예를 들어 표 6에서 궤적 $t2$ 는 질의 궤적과의 공간 거리, 시간 거리가 각각 20, 1400임을 알 수 있다. 셋째, 공간 거리와 시간 거리를 이용하여 시공간 거리를 계산한다. 계산된 시공간 거리는 $t2$ 의 경우 합

```

Spatio-temporal Similar Trajectory Search Algorithm (Q, DT, k) {
Input: 질의 궤적 Q, 데이터 궤적 집합 DT, 검색할 유사 궤적의 수 k
Output: 검색된 유사 궤적 집합 및 질의 궤적과의 시공간 거리
//Partial matching process based on common segment/s
시그니처 파일을 이용하여 한 개 이상의 동일한 도로 세그먼트를 갖는 후보 궤적 검색
Result1=Signature-based Trajectory Search(Q, DT);
검색된 후보 집합 Result1의 모든 궤적에 대한 공간 거리, 시간 거리 계산
후보집합 Result1의 모든 궤적에 대한 시공간 거리 계산
(A) STDist(Q, Result1) = (SD+ δ*TD)/2 // 합 접근 방법
OR
(B) STDist(Q, Result1) = ( (SD/δ)+1)*(TD+1) // 곱 접근 방법
Result1을 계산된 시공간 거리 순으로 정렬
k 개의 가장 유사한 궤적 반환 provides (k)
}
    
```

<그림 10> 시공간 유사 궤적 검색 알고리즘

접근 방법, 곱 접근 방법에 따라 각각 900, 1491 임을 알 수 있다. 넷째, 사용하고자 하는 시공간 거리 접근 방법에 따라 계산된 시공간 거리를 정렬한다. 예를 들어 합 접근 방법의 경우 정렬된 결과 집합은 <t2, t3, t1>가 된다. 반면에 곱 접근 방법의 경우 결과 집합은 <t3, t2, t1>가 된다. 마지막으로 검색된 결과 중 사용자로부터 입력받은 k개의 가장 유사한 궤적을 반환한다.

<표 6> 시공간 유사 궤적 검색의 예

Trajectories	R	D	W	TD	SD	AA	MA
Q	ER	Wed	1 st	-	-	-	-
t1	ER	Sat	2 nd	161	3000	3110	24462
t2	ER	Tue	1 st	20	1400	900	1491
t3	MR	Mon	4 th	103	0	1030	103
t4	MR	Fri	3 rd	-	-	-	-
t5	L	Wed	1 st	-	-	-	-
t6	N	Sun	5 th	-	-	-	-

4. 성능평가

제안하는 시공간 유사 궤적 검색 알고리즘의 성능 분석을 위하여 HP ML 150 G3 서버, 인텔 제온 3.0 GHz dual CPU, 2 GB 메모리, 250GB HDD, Windows 2003 서버 환경에서 Visual

Studio .Net 을 이용하여 구현하였다. 성능 분석을 위해 사용된 데이터는 샌프란시스코 만의 도로 네트워크를 Brinkhoff [22] 알고리즘을 사용하여 생성하였다. 도로 네트워크는 17만개의 노드와 22만개의 양방향 세그먼트로 구성되어 있으며, 생성된 이동 객체 궤적의 수는 2,000개, 이동 객체 궤적의 크기는 10~100까지의 다양한 크기로 생성하였다. 질의 궤적은 먼저, 제안하는 방법의 정확성을 증명하기 위해 생성된 데이터의 궤적 중에서 랜덤으로 1000개의 궤적을 추출하여 궤적의 10%~50%의 세그먼트를 대체하여 유사한 궤적을 생성하였고, 검색 성능 평가를 위해 데이터 궤적 생성과 동일한 방법으로 10~100개의 다양한 크기를 갖는 1,000개의 질의 궤적을 생성하였다. 또한 유사 궤적 검색 알고리즘을 위해 [19]의 알고리즘을 이용하여 시그니처를 생성하였으며 하나의 시그니처 크기는 180 바이트이다. 그림 11은 생성된 실제 질의 궤적 및 시그니처의 사용 예제를 나타낸다. 그림 11의 (a)에 나타난바와 같이 이동 객체 궤적을 변형하여 질의 궤적을 생성하였으며 붉은 색으로 표시된 부분은 대체된 구간을 나타낸다. 그림 11의 (b)는 이동 객체의 시그니처와 질의 궤적의 첫 번째 세그먼트의 시그니처를 나타내며 두 시그니처의 비트 AND 연산 결과가 질의 세그먼트의 시그니처와 같기 때문에 이동 객체는 검색 후보 집합에 포함되게 된다.

<p>이동 객체 궤적</p> <p>4 124993552 23327 23328 2005,2,10_THU_16:37 2005,2,10_THU_16:43</p> <p>4 124993408 23328 23323 2005,2,10_THU_16:43 2005,2,10_THU_16:56</p> <p>4 124993407 23323 23322 2005,2,10_THU_16:56 2005,2,10_THU_16:59</p> <p>4 124993402 23322 23316 2005,2,10_THU_16:59 2005,2,10_THU_17:7</p> <p>4 124993400 23316 23315 2005,2,10_THU_17:7 2005,2,10_THU_17:11</p> <p>4 124993482 23315 23744 2005,2,10_THU_17:11 2005,2,10_THU_17:21</p> <p>4 124993470 23744 23745 2005,2,10_THU_17:21 2005,2,10_THU_17:27</p> <p>4 124993466 23745 23771 2005,2,10_THU_17:27 2005,2,10_THU_17:34</p> <p>4 124993465 23771 23288 2005,2,10_THU_17:34 2005,2,10_THU_17:38</p> <p>4 124993463 23288 23772 2005,2,10_THU_17:38 2005,2,10_THU_17:44</p> <p>4 124993433 23772 23780 2005,2,10_THU_17:44 2005,2,10_THU_17:59</p>	<p>질의 궤적</p> <p>4 124993552 23327 23528 2005,2,10_THU_16:15 2005,2,10_THU_16:22</p> <p>4 124993408 23328 23323 2005,2,10_THU_16:22 2005,2,10_THU_16:38</p> <p>4 124993407 23323 23322 2005,2,10_THU_16:38 2005,2,10_THU_16:38</p> <p>4 124993403 23322 23321 2005,2,10_THU_16:38 2005,2,10_THU_16:44</p> <p>4 124993397 23321 23319 2005,2,10_THU_16:44 2005,2,10_THU_16:53</p> <p>4 124993371 23319 23318 2005,2,10_THU_16:53 2005,2,10_THU_16:59</p> <p>4 124993393 23318 23317 2005,2,10_THU_16:59 2005,2,10_THU_17:7</p> <p>4 124993401 23317 23314 2005,2,10_THU_17:7 2005,2,10_THU_17:12</p> <p>4 124993392 23314 23313 2005,2,10_THU_17:12 2005,2,10_THU_17:21</p> <p>4 124993469 23313 23745 2005,2,10_THU_17:21 2005,2,10_THU_17:27</p> <p>4 124993466 23745 23771 2005,2,10_THU_17:27 2005,2,10_THU_17:31</p> <p>4 124993465 23771 23288 2005,2,10_THU_17:31 2005,2,10_THU_17:39</p> <p>4 124993463 23288 23772 2005,2,10_THU_17:39 2005,2,10_THU_17:48</p> <p>4 124993433 23772 23780 2005,2,10_THU_17:48 2005,2,10_THU_17:57</p>
--	--

(a) 질의 궤적의 예

```

0040000000000000000100000091100408120080801000202
22008000000006080084800000000010002800d00004841
0040000000000920802800010104000080004000014000404
800000010000000000008040808020010400000000000818
000000004000180000000080000002008002821080008041
000100000000000400000000008000800000010100000010
20002020420000000020009000005000000000000000211
0000010000

00000000000000000000000008000040000008000000200
00000000000000000000000000000000000000000000040
0000000000000008000000000000000000000000000000
0000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000001
0000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000
0000000000
    
```

①이동 객체 궤적 시그니처

②질의 궤적의 첫 번째 세그먼트 시그니처

① & ② == ②

(b) 시그니처 사용의 예

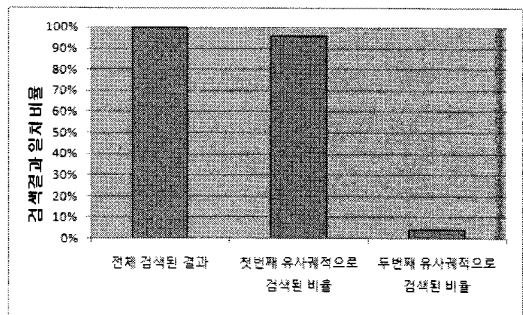
<그림 11> 생성된 질의 궤적 및 시그니처 사용 예제

성능 분석은 첫째, 제안하는 방법의 정확성을 증명하기 위해 추출된 질의 궤적에 대해 검색된 데이터의 비교를 수행하였고, 둘째, 시공간 거리의 계산을 위한 합 접근 방법과 곱 접근 방법의 비교를 수행하였고, 셋째, 도로 네트워크 환경에서 유클리디언 거리가 아닌 네트워크 거리에 기반하여 시공간 유사 궤적 검색의 관련 연구와 비교한다.

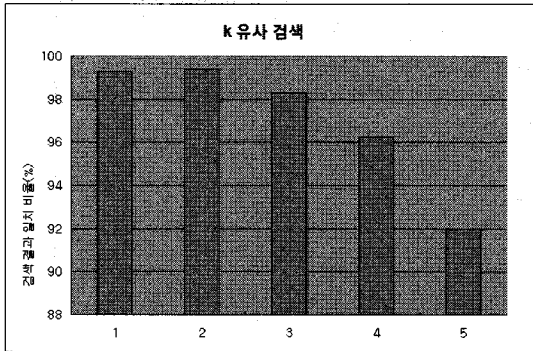
4.1 제안하는 방법의 정확성 평가

먼저, 제안하는 방법의 정확성을 증명하기 위해 데이터 궤적의 크기 및 세그먼트를 10%~50% 변형한 질의 궤적에 대한 검색 결과를 측정하였다. 그림 12는 검색된 데이터 궤적의 정확성 평가를 나타낸다. 그림에 나타난바와 같이 변형된 질의 궤적에 대한 유사 궤적 검색 결과 100% 원본 데이터를 검색하여 제안하는 방법이 정확함을 알 수 있다. 보다 자세히 살펴보면 원본 데이터가 질의 궤적에 대해 가장 유사한 궤적으로 검색된 비율은 약 96%이며,

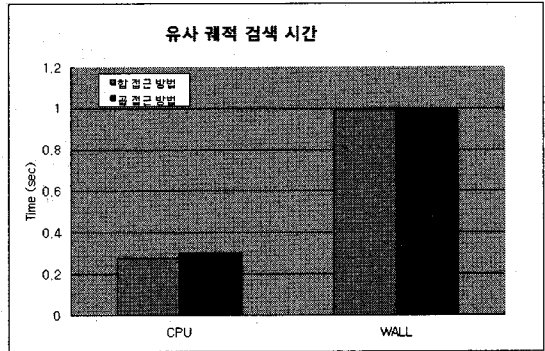
두 번째 유사 궤적으로 검색된 비율은 4%이다. 이러한 이유는 변형된 질의 궤적이 오히려 다른 데이터 궤적과 보다 유사해지면서 원본 데이터 궤적을 2번째 유사 궤적으로 검색하는 경우가 있기 때문이다. 결과적으로 질의 궤적에 대해 원본 데이터 궤적의 대부분이 첫 번째 유사궤적으로 검색되고 나머지의 경우도 두 번째 유사궤적으로 모두 검색되어 제안하는 방법이 정확함을 알 수 있다.



<그림 12> 제안하는 방법의 정확성 평가



<그림 13> 합 접근 방법과 곱 접근 방법의 검색 결과 일치 비율



<그림 14> 합 접근 방법과 곱 접근 방법의 검색 시간

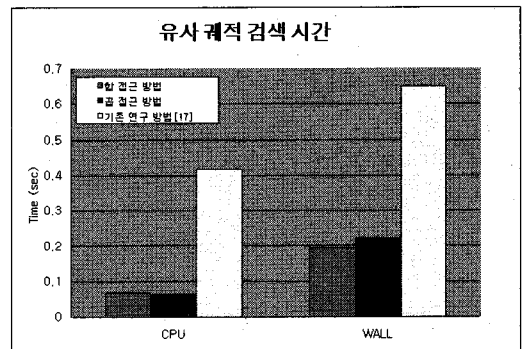
4.2 합 접근 방법과 곱 접근 방법의 비교

본 절에서는 시공간 유사도 측정 방법으로 제안한 두 접근 방법의 검색 결과를 비교하였다. 첫째, 그림 13은 두 접근 방법에 따른 검색 결과 중 일치하는 데이터의 비율을 나타낸다. 그림에서 k의 수가 증가할수록 일치하는 데이터의 비율은 낮아짐을 알 수 있다. K가 2일 경우 검색 결과 일치율은 99%이며, k가 5일 경우에는 92%임을 알 수 있다. 이러한 이유는 두 접근 방법에 따라 시간 거리 및 공간 거리에 대한 가중치의 적용으로 시공간 거리의 차이가 발생하기 때문이다. 둘째, 그림 14는 두 접근 방법에 따른 검색 시간을 나타낸다. 두 방법 모두 1초 미만의 적은 검색 시간이 소요됨을 알 수 있다. 이러한 이유는 시그니처 파일 기반 검색 기법을 사용하여 후보 결과 집합을 효율적으로 검색할 수 있기 때문이다.

4.3 제안하는 방법과 기존 연구와의 비교

본 절에서는 제안하는 두 가지 접근 방법과 도로 네트워크 환경에서의 기존 연구인 황정래 [17]의 방법과의 성능 비교를 수행하였다. 기존 연구와의 성능 비교를 위해 제안하는 방법을 질의 궤적을 구성하는 모든 세그먼트를 포함하는 데이터 궤적을 검색할 수 있도록 수정하여 성능 비교를 수행하였

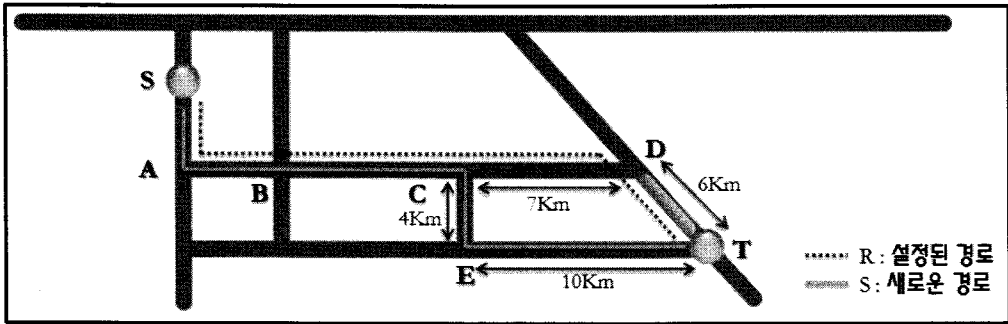
다. 그림 15는 제안하는 방법과 기존 연구와의 검색 시간을 나타낸다. CPU 시간의 경우, 제안하는 방법은 0.2초, 기존 연구는 0.4초의 시간이 소요되어 제안하는 방법이 우수한 성능을 나타냄을 알 수 있다. 디스크 접근 시간을 포함한 WALL 시간의 경우에도 마찬가지로 제안하는 방법이 기존 방법에 비해 3배 이상의 좋은 성능을 나타냄을 알 수 있다.



<그림 15> 제안하는 방법과 기존 연구와의 시공간 유사 궤적 검색 시간 비교

4.4 제안하는 방법의 활용 예제

현재 네비게이션 시스템이 추천하여 주는 경로는 거리상 최적의 경로이며, 도로의 지체/정체에 따라서 이것이 시간적인 측면에서 최적의 경로가 아닌 경우가 빈번히 발생한다. 또한, 현재의 도로 상태를



<그림 16> 제안하는 방법의 활용 예

알려주는 최첨단의 네비게이션 시스템의 경우에도 현재 시각에서의 도로의 지체/정체만을 반영하므로, 추천해준 경로가 1~2 시간 이후에 최적경로가 아닌 경우가 빈번히 발생한다. 본 절에서는 제안하는 방법은 과거의 이동 객체의 궤적 데이터를 데이터 마이닝 기법을 이용하여 분석하고 이를 기반으로 이러한 최적의 경로를 추천하기 위한 응용에 활용 할 수 있다. 예를 들면, 그림 16에서 출발지 S에서 목적지 T 까지 거리상 최적 경로로 선정된 $R = \langle (S,A), (A,B), (B,C), (C,D), (D,T) \rangle$ 의 (D,T) 구간이 시내 중심가이므로, 차량이 (D,T) 구간을 통과하는 시간이 출퇴근 시간일 경우는 이 구간에 심한 차량의 지체가 발생한다. 따라서 제안하는 유사 궤적 검색을 이용하여 질의 궤적과 유사한 궤적을 검색한 후, 이를 데이터마이닝 기법(classification, association)을 사용하여, 계절별, 요일별, 시간대 별로 도로구간의 평균 속도 및 최저 속도를 추출한다. 이를 통하여 (D,T)와 같은 지체/정체 구간을 발견하고 이를 대체하는 (C,E,T) 구간을 분석된 정보를 통해 검색하여 최적 경로인 $S = \langle (S,A), (A,B), (B,C), (C,E), (E,T) \rangle$ 를 설정할 수 있다.

5. 결 론

도로 네트워크 상에 존재하는 이동 객체의 유사 궤적 검색에 관련된 연구는 비교적 활발하게 수행된 반면, 이동 객체 궤적의 시간 및 공간 특성을 모두 고려한 연구는 거의 이루어지지 않았다. 이를 위

하여, 본 논문에서는 도로 네트워크와 같은 도로 네트워크 환경에서의 유사 궤적 검색을 위하여 유클리디언 거리 기반이 아닌 네트워크 거리에 기반한 새로운 유사도 측정 방법을 정의하였다. 첫째, 두 궤적 사이의 공간적인 특성에 기반하여 네트워크 거리를 이용한 공간 유사도 측정 방법을 정의하였고, 궤적의 시간을 시간 영역, 요일, 주의 세가지 데이터로 나누어 각각의 거리에 기반한 시간 유사도 측정 방법을 정의하였다. 아울러, 시공간 유사도 측정을 위해 정의된 두 시간 및 공간 거리를 이용하여 합 접근 방법 및 곱 접근 방법을 이용한 시공간 거리를 정의하였다. 둘째, 정의된 각 유사도 측정 방법을 이용하여 질의 궤적과 유사한 궤적을 검색할 수 있는 시공간 유사 궤적 검색 알고리즘을 제안하였고, 효과적인 검색을 위해 시그니처 기반 액세스 기법을 사용하였다. 마지막으로, 성능 분석을 통해 제안하는 방법이 기존 연구에 비해 우수한 성능을 나타냄을 증명하였다.

향후 연구로는 도로 네트워크 상의 자동차의 움직임과 같은 실제 이동 객체의 궤적 데이터를 이용하여 제안하는 알고리즘의 성능을 입증하는 것이다.

참고문헌

1. Betke, M., Gips, J., and Fleming, P., "The camera mouse: visual tracking of body features to provide computer access for people with severe disabilities,"

- IEEE Trans. Neural Syst. Rehabil. Eng., Vol.10 No.1, 2002, pp. 1-10.
2. Shimada, M., and Uehara, K., "Discovery of correlation from multi-stream of human motion," *Discovery Sci.*, 2000, pp. 290-294.
 3. Barbara, D., "Mobile computing and databases—a survey," *IEEE TKDE*, Vol.11 No.1, 1999, pp. 108-117.
 4. Roddick, J.F., and Hornsby, K., "An Updated Bibliography of Temporal, spatial and spatio-temporal data mining Research," *TSDM*, 2000, pp. 147-164.
 5. Chudova, D., Gaffney, S., Mjolsness, E., and Smyth, P., "Translation invariant mixture models for curve clustering," In *Proc. of 9th SIGKDD*, 2003, pp. 79-88.
 6. Arikan, O., and Forsyth, D., "Interactive motion generation from examples," In *Proc. of ACM SIGGRAPH*, 2002, pp. 483-490.
 7. Valdes-Perez, R.E., and Stone, C.A., "Systematic detection of subtle spatio-temporal patterns in time-lapse imaging," *Particle migrations, Bio-imaging*, Vol.6 No.2, 1998, pp. 71-78.
 8. Pfoser, D., Jensen, C.S., and Theodoridis, Y., "Novel Approaches in Query Processing for Moving Object Trajectories," In *Proc. of the 26th International Conference on Very Large Data Bases*, 2000, pp. 395-406.
 9. Vlachos, M., Kollios, G. and Gunopulos, D., "Discovering Similar Multidimensional Trajectories," In *Proc. of the 18th International Conference on Data Engineering*, 2002, pp. 673-684.
 10. Vlachos, M., Gunopulos, D., and Kollios, G., "Robust Similarity Measures for Mobile Object Trajectories," In *Proc. of the 13th International Workshop on Database and Expert Systems Applications*, 2002, pp. 721-728.
 11. Shim, C.B., and Chang, J.W., "Similar Sub-Trajectory Retrieval for Moving Objects in Spatio-temporal Databases," In *Proc. of The 7th East European Conference on Advances In Databases and Informations Systems*, 2003, pp. 308-322.
 12. Yanagisawa, Y., Akahani, J.I., and Satoh, T., "Shape-Based Similarity Query for Trajectory of Mobile Objects", In *Proc. of the 4th International Conference on Mobile Data Management*, 2003, pp. 63-77.
 13. Lin, B., and Su, J., "Shapes Based Trajectory Queries for Moving Objects," In *Proc. of 13th ACM GIS*, 2005, pp. 21-30.
 14. Zeinalipour-Yazti, D., Lin, S., and Gunopulos, D., "Distributed Spatio-Temporal Similarity Search," In *Proc. of 15th ACM CIKM*, 2006, pp. 14-23.
 15. Sakurai, Y., Yoshikawa, M., and Faloutsos, C., "FTW: Fast similarity search under the time warping distance," In *Proc. PODS*, 2005, pp. 326-337.
 16. Chen, L., Ozsu, M.T., and Oria, V., "Robust and fast similarity search for moving object trajectories," In *Proc. ACM SIGMOD*, 2005, pp. 491-502.
 17. Hwang, J.R., Kang, H.Y., and Li, K.J., "Spatio-temporal Similarity Analysis Between Trajectories on Road Networks," *ER Workshops*, 2005, pp. 280-289.
 18. Yanagisawa, Y., Akahani, J., and Satoh, T., "Shape-Based Similarity Query for

Trajectory of Mobile Objects”, In Proc. MDM 2003, pages 63–77, 2003.

19. Faloutsos, C., and Christodoulakis, S., “Signature Files: An Access Method for Documents and Its Analytical performance Evaluation,” ACM Tran. on Office Information Systems, Vol.2 No.4, 1984, pp. 267–288.

20. Zobel, J., Moffat, A., and Ramamohanarao, K., “Inverted Files versus Signature Files for Text Indexing,” ACM Tran. on Database Systems, Vol.23 No.4, 1998, pp. 453–490.

21. Chang, J.W., and Um, J.H., “An Efficient Indexing Scheme for Moving Objects’ Trajectories on Road Networks,” WAIM, 2006, pp. 13–25.

22. Brinkhoff, T., “A Framework for Generating Network-Based Moving Objects,” GeoInformatica, Vol.6 No.2, 2002, pp. 153–180.

김영창

2001년 전북대학교 컴퓨터공학과(공학사)
 2003년 전북대학교 대학원 컴퓨터공학과(공학석사)
 2004년~현재 전북대학교 대학원 컴퓨터공학과
 박사과정
 관심분야 : 데이터 마이닝, 공간 데이터베이스, 공간 색인 구조, 질의처리 알고리즘

라빈드라 비스타

2004년 Sikkim Manipal University, India,
 Information technology(공학사)
 2007년 전북대학교 컴퓨터공학과(공학석사)
 2007년~현재 전북대학교 컴퓨터공학과 박사과정
 관심분야 : 공간 데이터베이스, 질의처리 알고리즘

장재우

1984년 서울대학교 전자계산기공학과(공학사)
 1986년 한국과학기술원 전산학과(공학석사)
 1991년 한국과학기술원 전산학과(공학박사)
 1996년~1997년 Univ. of Minnesota, Visiting
 Scholar
 2003년~2004년 Penn State Univ., Visiting
 Scholar
 1991년~현재 전북대학교 컴퓨터공학과 교수
 관심분야 : 공간 네트워크 데이터베이스, 상황인식,
 하부저장구조