

도로 네트워크에서의 유사 궤적 클러스터링

백지행⁰, 원정임, 김상욱
한양대학교 정보통신학부

hanghang1010@hanmail.net⁰, jiwon@hanyang.ac.kr, wook@hanyang.ac.kr

Similar Trajectory Clustering on Road Networks

Ji-Haeng Baek⁰, Jung-Im Won, Sang-Wook Kim

School of Information and Communications, Hanyang University

요 약

본 논문에서는 도로 네트워크내의 이동 객체들을 대상으로 하는 효과적인 유사 궤적 검색 및 클러스터링 기법에 대하여 논한다. 이동 객체들 간의 유사도 측정을 위한 기존의 기법들은 대부분 유클리디안 공간 상의 궤적들을 대상으로 한다. 그러나 실제 응용에서 대부분의 이동 객체들은 도로 네트워크 공간 상에 존재하므로, 이러한 실제 상황을 반영하는 유사도 측정 방식이 요구된다. 본 논문에서는 각 이동 객체가 시간에 따라 지나간 도로 세그먼트들의 리스트를 궤적이라 정의하고, 이렇게 정의된 궤적들을 대상으로 하는 새로운 유사도 측정 함수를 제안한다. 제안된 유사도 측정 함수는 궤적을 이루는 도로 세그먼트의 길이와 식별자 정보를 이용한다. 제안된 유사도 측정 함수에 의하여 측정된 각 궤적 쌍 간의 유사도를 기반으로 전체 궤적들을 FastMap을 이용하여 k차원 공간상의 점들로 사상하고, 이들을 k-medoids 방식을 이용하여 클러스터링 한다. 구성된 클러스터와 연관된 사용자 정보, 도로 정보 등을 함께 사용자에게 제공하는 활용 예를 제시함으로써 제안된 기법이 실제 응용에 유용하게 사용될 수 있음을 보인다.

1. 서론

최근, 위성 기반 시스템의 발전과 모바일 기기와 같은 휴대용 이동 장비의 보편화로 인하여 텔레매틱스(telematics) 기술이 크게 발전하고 있다. 텔레매틱스 시스템을 도입하여 실시간 교통 정보 제공, 이동 경로 안내, 차량 사고 알림 등의 다양한 서비스가 제공되고 있다. 이들은 일정 시간 간격마다 위성 내의 GPS(global positioning system)를 이용하여 측정된 이동 객체의 궤적 정보를 기반으로 한다. 최근, 이들 궤적 정보를 효율적으로 저장 및 관리하는 기법들에 관한 많은 연구들이 보고되고 있다[6, 8, 14, 15]. 또한, 몇몇 연구에서는 주어진 이동 객체의 궤적과 유사한 궤적을 검색 또는 클러스터링하고, 이를 도로 정보 및 사용자 정보 등과 연계하여 분석하려는 시도가 활발히 진행되고 있다[5, 6, 7].

이들 대부분의 연구는 유클리디안(Euclidean) 공간 상에서 2차원의 공간 좌표 (x,y)의 연속으로 표현되는 이동 객체의 궤적 정보를 대상으로 하며, 유사 궤적을 검색하기 위한 유사도 측정 방식으로 공간 좌표 상의 거리 측정 방식인 유클리디안 거리를 사용한다. 그러나 실제 응용에서 대부분의 이동 객체들은 도로 네트워크(road network) 공간 상에 존재하며, 이동 객체의 공간 정보를 파악하기 위하여는 1차원의 도로 정보가 사용자에게 보다 직관적이고, 유용한 정보를 제시할 수 있다. 예를 들어, 이동 객체가 '한양대역'을 지나고 있다면 이를 (동경

360도40분, 북위56도34분)과 같이 2차원 공간 좌표로 표현하는 것이 아니라 '왕십리길'로 표현하는 것이 더 실제계에 부합하는 표현 방식이라 할 수 있다. 최근, 제한된 도로 네트워크 공간에서의 움직임을 갖는 이동 객체의 궤적 정보를 효과적으로 표현, 저장, 인덱싱 하고자 하는 몇몇 연구가 시도된바 있다[6, 8, 14, 16, 17]. 그러나 유사 궤적 검색 및 클러스터링 기법에 관한 연구는 아직 미흡한 상태이다.

본 논문에서는 도로 네트워크 공간상에서 이동 객체들이 시간 흐름에 따라 움직인 대용량 궤적 정보를 대상으로 주어진 길의 궤적과 유사한 궤적 및 궤적 클러스터를 효율적으로 검색하기 위한 기법에 대하여 논한다. 먼저, 이동 객체의 궤적이 2차원 공간 좌표 (x, y)의 연속으로 표현되는 유클리디안 공간과는 달리, 이동 객체가 지나온 도로 세그먼트들의 연속으로 표현한다. 이때, 각 도로 세그먼트의 정보로는 식별자와 길이가 사용된다.

다음, 도로 세그먼트의 식별자 및 길이 정보를 이용한 유사도 측정 방식을 제안한다. 기존의 유클리디안 공간 상에서의 두 궤적간의 유사도 측정 방식을 도로 네트워크 공간 상에 적용하는 것은 적합하지 않다. 예를 들어, A라는 고가도로로 움직인 이동 객체의 궤적과 해당 고가도로의 밑으로 이동하는 이동 객체의 궤적을 유클리디안 거리 함수를 이용하여 유사도를 측정하는 경우 두 궤적은 서로 다른 도로를 지나고 있음에도 불구하고

유사한 것으로 판정되기 때문이다.

또한, 본 논문에서는 측정된 궤적간의 유사도 정보를 기반으로 기존의 FastMap[11]과 k-medoids[12, 13] 기법을 이용하여 궤적들을 클러스터링 한다. 구성된 클러스터와 연관된 사용자 정보, 도로 정보 등을 함께 사용자에게 제공하는 예를 제시함으로써 제안된 기법이 실제 응용에 유용하게 사용될 수 있음을 보인다.

본 논문의 구성은 다음과 같다. 제 2장에서는 유사 궤적을 검색하기 위한 기존의 유사도 측정 방식에 대하여 설명하고, 이들 방식을 도로 네트워크에 적용했을 때의 문제점에 대하여 설명한다. 제 3장에서는 본 논문에서 제안하는 유사도 측정 방식과 이를 이용한 클러스터링 방식에 대하여 설명하고, 실제 응용에서의 분석된 클러스터 정보의 활용 예를 보인다. 마지막으로, 제 4장에서는 본 논문의 결론을 내리고, 향후 연구를 제시한다.

2. 관련 연구

유클리디안 공간상에서 이동 객체 궤적들 간의 유사도 측정을 위하여 EU(Euclidean distance), DTW(dynamic time warping distance), ERP(edit distance with real penalty), LCSS(longest common sub-sequences), EDR(edit distance in real sequence) 등의 거리 함수를 이용한 기법들이 제안되어 있다[1, 2, 3, 4].

EU 방식은 길이가 같은 두 궤적이 주어졌을 때, 궤적을 구성하는 k차원 시공간 좌표들 간의 유클리디안 거리를 구하는 방식이다. 이 방식은 비교하는 두 궤적의 길이가 동일해야 한다는 제약이 있다. 실제 응용에서는 궤적들의 길이가 동일하지 않기 때문에 이 방법은 적합하지 않다. DTW 방식은 서로 다른 길이를 갖는 궤적들 간의 유사도를 측정하기 위하여 궤적내의 특정 시공간 좌표 값을 임의의 수만큼 반복시키는 것을 허용하는 방식이다. ERP 방식은 유사 궤적 내에 갭을 허용하며, 길이가 서로 다른 두 궤적간의 유사도 측정이 가능하지만 잡음에 매우 민감하다는 단점이 있다. EU, DTW, ERP 함수는 모두 잡음에 민감한 유사도 측정 방식이므로, 궤적의 획득 혹은 표현 과정에서 잡음이 발생할 가능성이 높은 실제의 궤적들을 대상으로 할 경우 정확도가 낮다는 단점을 갖는다.

이러한 문제점을 해결한 방식으로 LCSS 방식과 EDR 방식을 들 수 있다. 이들 방식은 잡음에 의한 민감도를 감소시키기 위하여 유사 궤적내의 시공간 좌표 값의 차이가 주어진 허용치보다 작으면 두 좌표가 매칭된 것으로 간주하는 방식이다. 그러나 LCSS 방식은 궤적 내에 갭을 허용하지 않는 유사도 측정 방식으로 정확도가 낮다는 단점을 갖는다. EDR 방식은 유사도 측정을 위하여 에디트 거리(edit distance)를 사용하며, LCSS와 달리 궤적내의 갭을 허용하는 방식이다.

이들 방식들은 모두 유클리디안 공간을 기반으로 하

므로 본 논문에서 대상으로 하는 도로 네트워크 공간 상에서의 궤적들 간의 유사도 측정에는 적합하지 않다. 도로 네트워크 공간 상에서의 궤적은 이동 객체가 거처간 도로 세그먼트들의 연속으로 표현되며, 이 경우 연속적으로 동일 도로 세그먼트가 반복적으로 궤적 내에 출현하는 경우는 발생하지 않는다. 따라서 반복을 허용하는 DTW나 ERP 방식은 적용할 수가 없다. 또한, LCSS나 EDR 방식은 궤적을 구성하는 도로 세그먼트의 식별자에 대하여 유사도 측정이 가능하나, 서브 궤적에 대한 유사도 측정 과정을 반복적으로 수행해야하므로 이로 인한 성능 저하 문제가 발생할 수 있다.

3. 제안하는 기법

본 논문에서는 이동 객체의 궤적을 $T_i = \{(S_1, L_1), \dots, (S_n, L_n)\}$ 으로 표현한다. 여기서, T_i 은 궤적의 식별자이고, S_j ($1 \leq j \leq n$)는 세그먼트의 식별자, L_j 는 세그먼트의 길이 이룰 나타낸다. 또한, 궤적 정보와 함께 사용자 정보, 도로 정보 등을 부가 정보로 함께 저장, 관리한다. 그림 1은 궤적 표현의 예를 나타낸 것이다.

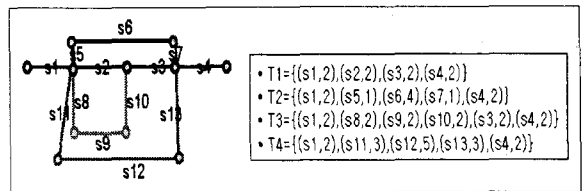


그림 1. 궤적 표현의 예.

본 논문에서는 데이터베이스에 저장된 궤적 정보를 대상으로 클러스터링을 수행하기 전에 먼저 궤적들 간의 유사도를 측정하기 위한 함수를 제안한다. 다음, 모든 궤적 쌍 간의 유사도와 FastMap[11] 방식을 이용하여 모든 궤적들을 k차원의 점들로 매핑한 후, k-medoids[12, 13] 방식을 이용하여 클러스터링을 수행한다. 마지막으로, 주어진 길의 궤적에 대하여 해당 클러스터를 신속하게 검색하기 위한 길의 처리 기법을 제안한다. 또한, 검색된 클러스터와 연관된 사용자 정보, 도로 정보 등을 함께 사용자에게 제공하는 활용 예를 제시함으로써 제안된 기법이 실제 응용에 유용하게 사용될 수 있음을 보인다.

3.1 유사도 측정 함수

제안하는 기법에서 궤적은 문자열로 표현되는 세그먼트 식별자들의 리스트로 구성된다. 주어진 두 궤적간의 유사도 측정을 위하여 문자열간의 거리 함수로 많이 사용되는 ED(edit distance) 함수를 이용할 수 있다. 그러나 비교되는 두 궤적의 세그먼트 개수가 서로 다른 경우 유사도는 세그먼트 개수가 많은 궤적에 의하여 영향을 받는다. 따라서 ED 함수에 의하여 유사 궤적을 검색하는 것은 적합하지 않다. 본 논문에서는 궤적 T_i 와 T_j 간의 식

(1)의 유사도 측정 함수 DSN(disimilarity with number)을 제안한다.

$$DSM(T_i, T_j) = \frac{T_i \text{와 } T_j \text{의 비공통 세그먼트의 개수}}{T_i \text{의 세그먼트 개수} + T_j \text{의 세그먼트 개수}} \quad (1)$$

예를 들어, 그림 1의 궤적들에 대하여 DSN 함수에 의한 유사도 값을 구하면 다음과 같다.

$$\begin{aligned} DSN(T1, T2) &= 0.56, & DSN(T1, T3) &= 0.4, \\ DSN(T1, T4) &= 0.56, & DSN(T2, T3) &= 0.64, \\ DSN(T2, T4) &= 0.6, & DSN(T3, T4) &= 0.64 \end{aligned}$$

T1과 비교되는 T2, T3, T4에 대한 ED 함수에 의한 유사도는 모두 동일한 3이라는 값이 나온다. 반면, 제안된 DSN 함수에 의한 값은 위에 제시된 바와 같이 T1에 대해 T3이 가장 유사한 값을 보인다. 실제 T1과 T3의 공통된 세그먼트가 더 많기 때문에 ED의 값보다 DSN의 값이 더 신뢰성이 있다. 이는 DSN 함수는 두 궤적의 세그먼트 개수의 합을 이용하여 두 궤적의 거리를 계산하기 때문이다. 즉, 유사도 계산 시 ED 함수에서와 같이 두 궤적 중에서 세그먼트의 개수가 많은 임의의 한 궤적에 의한 영향을 감소시킬 수 있다. 그러나 DSN 함수 역시 다음과 같은 문제점을 갖는다.

- 1) 세그먼트 식별자의 매치 여부와 개수만을 가지고 유사도를 측정하므로 두 궤적의 실제 거리상의 근접 정도를 알 수 없다. 예를 들어, DSN 함수에 의하여 궤적 T1과 T2의 유사도와 궤적 T1과 T4의 유사도가 같은 값을 갖는다. 이는 T1과 T2의 비공통 세그먼트 개수와 T1과 T4의 비공통 세그먼트의 개수가 동일하기 때문이다. 그러나 그림 1에 보인바와 같이 T4의 궤적보다 T2의 궤적이 T1에 더 근접해 있으므로, T1과 T2가 더 유사하다고 할 수 있다.
- 2) 동일한 도로라 하더라도 세그먼트를 나누는 기준에 따라 궤적에 포함된 세그먼트의 개수가 달라질 수 있다. 제안된 DSN 함수는 세그먼트의 개수에 영향을 받기 때문에 하나의 세그먼트가 다수의 세그먼트로 분할되어 표현될 경우, 유사도 값이 전혀 다른 값을 갖게 된다. 예를 들어, 그림 1에서 두 궤적 T1과 T2의 $DSN(T1, T2) = 0.56$ 이다. 만약, 궤적 T2의 세그먼트 s6을 3개의 세그먼트로 분할했다면 $DSN(T1, T2) = 0.64$ 가 된다.

따라서 두 궤적의 유사도를 세그먼트의 개수만을 이용하여 판단할 수 없다. 본 논문에서는 이러한 문제점을 해결하기 위하여 세그먼트의 길이를 이용한 식 (2)의 유사도 측정 함수 DSL(dissimilarity with length)를 제안한다.

$$DSL(T_i, T_j) = \frac{T_i \text{와 } T_j \text{의 비공통 세그먼트의 길이의 합}}{T_i \text{의 세그먼트 길이의 합} + T_j \text{의 세그먼트 길이의 합}} \quad (2)$$

그림 1의 궤적에 대하여 DSL 함수에 의한 유사도 값을 구하면 다음과 같다.

$$\begin{aligned} DSL(T1, T2) &= 0.56, & DSL(T1, T3) &= 0.4, \\ DSL(T1, T4) &= 0.65, & DSL(T2, T3) &= 0.59, \\ DSL(T2, T4) &= 0.68, & DSL(T3, T4) &= 0.7 \end{aligned}$$

DSN 함수에 의한 $DSN(T1, T2)$ 과 $DSN(T1, T4)$ 의 유사도 값이 동일한 반면에, DSL 함수에 의한 $DSL(T1, T2)$ 과 $DSL(T1, T4)$ 의 유사도 값은 다름을 알 수 있다. 즉, DSL 함수에 의하여 궤적 T2가 궤적 T4보다 궤적 T1과 더 유사함을 파악할 수 있다. 제안된 DSL 함수는 두 궤적 간에 공통 세그먼트의 개수가 많고 세그먼트의 길이의 차이가 적은 궤적들을 보다 유사하다고 판단한다.

또한, 제안된 방식은 세그먼트를 나누는 기준에 따른 세그먼트 개수의 변화에 영향을 받지 않는다. 예를 들면, 궤적 T1, T2의 유사도 측정 시 s6의 세그먼트를 3개로 분할할 경우에 분할된 세 개의 세그먼트의 길이를 합한 전체의 길이는 4로 분할 전과 길이면에서는 변함이 없다. 따라서 분할 전과 분할 후의 $DSL(T1, T2)$ 의 값은 0.56으로 동일하다.

3.2 클러스터링 방법

제 3.1절에서 제안한 유사도 측정 함수를 이용하여 주어진 길의 궤적과 유사한 궤적을 가지는 이동 객체를 검색할 수 있다. 그러나 본 연구의 목적은 검색된 유사 궤적을 클러스터링하여 보다 유용한 정보를 사용자에게 제공하는 것이다.

궤적 클러스터링이란 궤적간의 유사도를 이용하여 그룹화하는 것을 말하며, 일반적인 클러스터링 방법에서는 클러스터의 무게 중심이라 할 수 있는 중심점의 반복적인 변경에 의해 클러스터를 구성한다. 궤적간의 상대적인 거리로 클러스터를 구성한다면 중심점이라는 기준이 모호해진다. 이러한 문제점을 해결하기 위하여 본 연구에서는 FastMap[11]을 이용하여 각 궤적을 k차원 공간 상의 한 점으로 표현한 후, 전체 궤적들과 대응되는 이 점들에 대하여 클러스터링을 수행한다.

이때, 서로 다른 길이를 갖는 궤적들을 하나의 차원으로 매핑시키기 위하여 본 연구에서는 제 3.1절에서 제안된 유사도 측정 함수 DSL에 의해 측정된 두 궤적간의 유사도 값을 이용한다. 여기서, 측정된 DSL의 값은 두 궤적간의 거리를 의미한다.

FastMap은 n개의 객체들과 객체간 거리 함수가 주어졌을 때, 이 객체들을 k차원 상의 점으로 매핑하는 기법이다. FastMap을 이용하면, 측정된 두 객체간의 거리를 이용하여 객적을 쉽게 k차원 공간 상의 점으로 매핑할 수 있다.

FastMap을 이용하여 객적을 k차원의 점으로 변환 방법은 다음과 같다.

1. DSL의 값 중 가장 큰 값을 갖는 두 객적을 찾는다. 이 두 객적을 T_{maxa} , T_{maxb} 라 한다.
2. 각 객적들은 다음 식에 의해 T_{maxa} , T_{maxb} 를 기준으로 하는 T_i ($1 \leq i \leq n$) 점으로 매핑된다.

$$T_i = \frac{DSL(T_{maxa}, T_i)^2 + DSL(T_{maxa}, T_{maxb})^2 - DSL(T_{maxb}, T_i)^2}{2 \times DSL(T_{maxa}, T_{maxb})}$$
3. 2에 의해 매핑된 T_i 점들의 새로운 거리차를 아래 식에 의해 재계산 한다.

$$DSL(T_i, T_j) = \sqrt{DSL(T_i, T_j)^2 - (T_i - T_j)^2}$$
4. 위의 1, 2, 3의 과정을 반복하며 매핑시킬 차원을 확장한다.

FastMap을 이용하여 객적을 k차원의 한 점으로 변환한 후, 본 연구에서는 k-medoids 방식[12, 13]을 이용하여 변환된 k차원 점들을 대상으로 클러스터링을 수행한다. k-medoids 방식은 각 클러스터에서 대표 객체(medoids)를 임의로 찾음으로써 n개의 객체 중에서 k개의 군집을 찾는 것이다. 클러스터의 대표 객체란 그 군집에 속하는 객체 중 다른 객체들과의 평균(또는 전체) 거리가 최소가 되는 객체를 말한다. k-medoids 클러스터링 방법은 객체들을 주어진 수의 클러스터로 구분하는데, 클러스터 내의 각 객체와 대표 객체와의 거리의 총합을 최소로 하는 방법이다. 이 방법은 k-means 방식에 비하여 잡음에 덜 민감하다는 장점을 갖기 때문에 본 연구에서는 이 방법을 채택하였다.

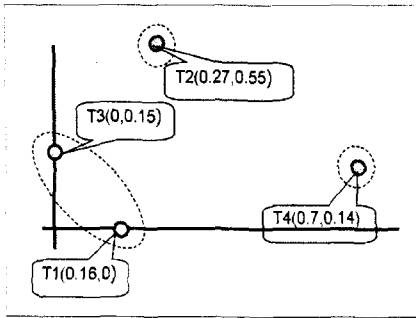


그림 2. 클러스터링 예.

그림 2는 FastMap을 이용하여 그림 1의 객적을 2차원의 점으로 매핑한 후, 이를 k-medoids 방식에 의하여 3개의 클러스터로 만드는 예를 보인다. 클러스터링 된 결과를 보면, 유사도가 가장 높은 T1과 T3가 같은 클러스터로 그룹화되어 있는 것을 확인할 수 있다.

본 연구에서는 클러스터링 과정을 통하여 얻어진 클러스터에 대한 정보를 각 클러스터 C에 대하여 세그먼트 요약 정보 $S = \langle (SegList_1, Weight_1), \dots, (SegList_n, Weight_n) \rangle$ 와 사용자 요약 정보 $U = \{User_1, \dots, User_m\}$ 을 구성하여 함께 저장 관리한다. 여기서, $SegList_n$ 는 해당 클러스터 내에 포함되는 각 객적의 세그먼트 리스트를 의미하며, $Weight_n$ 는 클러스터 내에서 해당 세그먼트 리스트의 발생 빈도를 의미하며 식 (3)에 의하여 얻어진다.

$$\text{빈도수} = \frac{\text{클러스터 내 해당 세그먼트 리스트가 포함된 객적의 개수}}{\text{클러스터 내 객적의 총 개수}} \quad (3)$$

예를 들어, 그림 1의 각 객적에 대하여 클러스터링 과정을 수행하여 얻어진 $C1 = \{T1, T3\}$ 이고 각 객적 사용자가 각각 user1, user2라면, 세그먼트 요약 정보 $S = \langle (s1, s3, s4), 1 \rangle, \langle (s2, s8, s9, s10), 0.5 \rangle$, 사용자 요약 정보 $U = \{user1, user2\}$ 이다.

3.3 질의 처리

본 절에서는 제 3.2절에서 구성된 클러스터의 세그먼트 요약 정보와 사용자 요약 정보를 이용한 효율적인 질의 처리 기법을 제안한다. 제안된 기법에서는 주어진 질의 객적 내에 포함된 각 세그먼트를 각 클러스터의 세그먼트 요약 정보내의 해당 세그먼트의 가중치 값과 비교한다.

예를 들어, 그림 1로부터 구성된 클러스터의 세그먼트 요약 정보에 대하여 질의 객적 $Q = \{s1, s2, s10, s9, s8, s5\}$ 가 주어지면 Q를 구성하는 각 세그먼트에 대하여 클러스터의 세그먼트 요약 정보를 검색한다. 각 세그먼트와 비교된 각 클러스터의 세그먼트 요약 정보내의 해당 세그먼트의 가중치 값들의 합을 구하고, 그 중 가장 높은 가중치 값을 갖는 클러스터를 검색한다. 예에서 Q내의 세그먼트 s1, s2, s10, s9, s8는 클러스터 C1에 존재하므로 이때의 s1, s2, s10, s9, s8의 가중치를 모두 합하면 $1+0.5+0.5+0.5+0.5=3$ 이 된다. 다음 Q내의 세그먼트 s1, s5는 클러스터 C2에 존재하므로 가중치의 합은 $1+1=2$ 이고, 세그먼트 s1은 클러스터 C3에 존재하므로 가중치의 합은 1이다. 질의 객적 Q와 매핑되는 클러스터는 가중치 값이 가장 큰 클러스터 C1이 된다. 따라서 주어진 질의 객적에 대하여 클러스터를 재구성하는 추가적인 오버헤드 없이 질의 객적과 매치되는 클러스터를 검색할 수 있다.

다음, 검색된 클러스터와 연관된 사용자 정보와 도로 정보 등을 사용자에게 함께 제시한다. 예를 들어, 클러스터 C1에는 서울 지역에서 유사한 객적으로 움직인 사용자 A, B, C가 존재하고, 클러스터 C2에는 부산 지역에서

유사 궤적으로 움직인 사용자 A, B가 존재한다고 가정한다면, 우리는 사용자 C가 부산 지역을 방문할 경우 클러스터 C1에 속한 사용자 A와 B가 부산 지역에서 이동한 경로를 사용자 C에게 추천할 수 있다.

4. 결론

본 논문에서는 도로 네트워크에서 움직이는 이동 객체들의 대용량 궤적 데이터를 대상으로 하는 유사 궤적 검색 및 클러스터링 기법에 관하여 논하였다. 이를 위하여 우선 도로 세그먼트의 식별자와 길이 정보를 이용한 새로운 유사도 측정 함수를 제안하였으며, 이를 기반으로 하는 효율적인 클러스터링 기법을 제안하였다. 또한, 구성된 클러스터에 대한 세그먼트 요약 정보를 활용하여 주어진 질의 궤적과 매칭되는 클러스터를 효율적으로 검색할 수 있음을 보였다.

향후 연구로서 다양한 실험을 통한 성능 평가를 이용하여 제안된 기법의 우수성을 규명하는 연구를 추진하고 있다.

감사의 글

본 논문은 제주대학교를 통한 정보통신부 및 정보통신진흥원의 대학 IT연구센터 지원사업의 지원을 받았습니다. (IITA-2005-C1090-0502-0009)

참고 문헌

[1] L. Chen, M. Ozsu, and V. Oria, "Robust and Fast Similarity Search for Moving Object Trajectories," In *Proc. Int'l. Conf. on Management of Data*, ACM SIGMOD, pp. 491-502, 2005.

[2] M. Vlachos, G. Kollios, and D. Gunopulos, "Dis-covering Similar Multidimensional Trajectories," In *Proc. Int'l. Conf. on Data Engineering*, IEEE ICDE, pp. 673-684, 2002.

[3] L. Chen and R. Ng, "On the Marriage of Lp-norms and Edit Distance," In *Proc. Int'l. Conf. on Very Large Data Bases*, VLDB, pp. 1040-1049, 2004.

[4] S. Chu et al., "Iterative Deepening Dynamic Time Warping for Time Series," In *Proc. SIAM Int'l. Conf. on Data Mining*, 2002.

[5] Y. Yanagisawa, J. Akahani, and T. Satoh, "Shape-Based Similarity Query for Trajectory of Mobile Objects," In *Proc. Int'l. Conf. on Mobile Data Management*, pp. 63-77, 2003.

[6] S. Brakatsoulas, D. Pfoser, and N. Tryfona, "Modeling, Storing, and Mining Moving Object Databases," In *Proc. Int'l. Symp. on Database Engineering and Applications*, pp. 68-77, 2004.

[7] 황 정래, 강 해영, 이 기준, "시공간 유사성을 이용 도로 네트워크 상의 유사한 궤적 검색," *정보처리학회 논문지*, Vol. 13-D, No. 3, pp. 337-346, 2006년.

[8] D. Pfoser and C. Jensen, "Indexing of Network Constrained Moving Objects," In *Proc. Int'l. Symp. on Advances in Geographic Information Systems*, ACMGIS, pp. 25-32, 2003.

[9] D. Papadias et al., "Query Processing in Spatial Network Databases," In *Proc. Int'l. Conf. on Very Large Data Bases*, VLDB, pp. 802-813, 2003.

[10] K.-S. Kim et al., "Fast Indexing and Updating Method for Moving Objects on Road Networks," In *Proc. IEEE Int'l. Conf. on Web Information Systems Engineering*, pp. 34-42, 2003.

[11] C. Faloutsos and K. Lin, "Fastmap: A Fast Algorithm for Indexing, Data-Mining, and Visualization of Traditional and Multimedia Datasets," In *Proc. Int'l. Conf. on Management of Data*, ACM SIGMOD, pp. 163-174, 1995.

[12] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, Academic Press, 2001.

[13] A. Reynolds, G. Richards, and V. Rayward-Smith, "The Application of K-Medoids and PAM to the Clustering of Rules," *Intelligent Data Engineering and Automated Learning*, pp. 173-178, 2004.

[14] V. Almeida and R. Güting, "Indexing the Trajectories of Moving Objects in Networks," *Geoinformatica*, Vol. 9, No. 1, pp. 33-60, 2005.

[15] O. Wolfson et al., "Moving Object Databases: Issues and Solutions," In *Proc. Int'l. Conf. on Scientific and Statistical Database Management*, SSDBM, pp. 111-122, 1998.

[16] M. Kolahdouzan and C. Shahabi, "Voronoi-Based K-Nearest Neighbor Search for Spatial Network Databases," In *Proc. Int'l. Conf. on Very Large Data Bases*, VLDB, pp. 840-851, 2004.

[17] X. Huang et al., "The Islands Approach to Nearest Neighbor Querying in Spatial Networks," In *Proc. Int'l. Symp. on Spatial and Temporal Databases*, SSTD, pp. 73-90, 2005.