

이산 프레셰 거리 척도를 이용한 궤적 유사도 고속계산 휴리스틱 알고리즘

Fast Heuristic Algorithm for Similarity of Trajectories Using Discrete Frchet Distance Measure

저자 (Authors)	박진관, 김태용, 박보국, 조환규 Jinkwan Park, Taeyong Kim, Bokuk Park, Hwan-Gue Cho
출처 (Source)	정보과학회 컴퓨팅의 실제 논문지 22(4) , 2016.4, 189-194(6 pages) KIISE Transactions on Computing Practices 22(4) , 2016.4, 189-194(6 pages)
발행처 (Publisher)	한국정보과학회 The Korean Institute of Information Scientists and Engineers
URL	http://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE06648726
APA Style	박진관, 김태용, 박보국, 조환규 (2016). 이산 프레셰 거리 척도를 이용한 궤적 유사도 고속계산 휴리스틱 알고리즘. 정보과학회 컴퓨팅의 실제 논문지, 22(4), 189-194
이용정보 (Accessed)	아주대학교 202.30.7.*** 2021/10/11 16:28 (KST)

저작권 안내

DBpia에서 제공되는 모든 저작물의 저작권은 원저작자에게 있으며, 누리미디어는 각 저작물의 내용을 보증하거나 책임을 지지 않습니다. 그리고 DBpia에서 제공되는 저작물은 DBpia와 구독계약을 체결한 기관소속 이용자 혹은 해당 저작물의 개별 구매자가 비영리적으로만 이용할 수 있습니다. 그러므로 이에 위반하여 DBpia에서 제공되는 저작물을 복제, 전송 등의 방법으로 무단 이용하는 경우 관련 법령에 따라 민, 형사상의 책임을 질 수 있습니다.

Copyright Information

Copyright of all literary works provided by DBpia belongs to the copyright holder(s) and Nurimedia does not guarantee contents of the literary work or assume responsibility for the same. In addition, the literary works provided by DBpia may only be used by the users affiliated to the institutions which executed a subscription agreement with DBpia or the individual purchasers of the literary work(s) for non-commercial purposes. Therefore, any person who illegally uses the literary works provided by DBpia by means of reproduction or transmission shall assume civil and criminal responsibility according to applicable laws and regulations.

이산 프레셰 거리 척도를 이용한 궤적 유사도 고속계산 휴리스틱 알고리즘 (Fast Heuristic Algorithm for Similarity of Trajectories Using Discrete Fréchet Distance Measure)

박진관^{*} 김태용^{**} 박보국^{*} 조환규^{***}
(Jinkwan Park) (Taeyong Kim) (Bokuk Park) (Hwan-Gue Cho)

요약 궤적은 이동체가 움직인 경로이다. IT 기술의 성장은 GPS와 같은 위치 측정 장치를 통해 다양한 이동체의 궤적 데이터를 수집할 수 있게 하였다. 이동체의 궤적은 지리정보시스템(GIS)을 포함한 다양한 연구 분야에서 사용된다. 지리정보시스템 분야에서는 차량의 궤적 데이터를 이용한 전자 도로 지도 생성 시도가 많이 이루어져왔다. 이 목표를 이루기 위해서는 같은 도로 상의 궤적들을 모으는 방법이 필요하다. 흔히 프레셰 거리(Fréchet distance)가 궤적 쌍의 거리를 측정하는데 사용된다. 하지만 프레셰 거리는 대량의 궤적들에 대해서는 계산 시간의 소모가 심하다. 본 논문에서는 궤적들의 인접성 여부를 이산 프레셰 거리를 통해 빠르게 구분하는 휴리스틱 알고리즘을 제안한다. 이 알고리즘은 계산되는 거리의 정확도를 낮추는 대신 계산 속도를 높였다. 실험 결과, 제안 방법은 이산 프레셰 거리 대비 95%의 정확도와 최하 65%의 계산 감소율로 거리가 10m 이내인 궤적들을 구분할 수 있었다.

키워드: 차량 GPS 데이터, 궤적 거리, 프레셰 거리, 궤적 유사도

Abstract A trajectory is the motion path of a moving object. The advances in IT have made it possible to collect an immeasurable amount of various type of trajectory data from a moving object using location detection devices like GPS. The trajectories of moving objects are widely used in many different fields of research, including the geographic information system (GIS) field. In the GIS field, several attempts have been made to automatically generate digital maps of roads by using the vehicle trajectory data. To achieve this goal, the method to cluster the trajectories on the same road is needed. Usually, the Fréchet distance measure is used to calculate the distance between a pair of trajectories. However, the Fréchet distance measure requires prolonged calculation time for a large amount of trajectories. In this paper, we presented a fast heuristic algorithm to distinguish whether the trajectories are in close distance or not using the discrete Fréchet distance measure. This algorithm trades the accuracy of the resulting distance with decreased calculation time. By experiments, we

· 이 논문은 2014년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. NRF-2014R1A1A2054687)

· 이 논문은 2015 한국컴퓨터종합학술대회에서 '이산 프레셰 거리 척도를 이용한 궤적 유사도 고속계산 휴리스틱'의 제목으로 발표된 논문을 확장한 것임

^{*} 비회원 : 부산대학교 전기전자컴퓨터공학과
jinkwan@pusan.ac.kr
limited10@pusan.ac.kr

^{**} 학생회원 : 부산대학교 전기전자컴퓨터공학과
ktyong22@pusan.ac.kr

^{***} 종신회원 : 부산대학교 전기전자컴퓨터공학과 교수
(Pusan National Univ.)
hgcho@pusan.ac.kr
(Corresponding author임)

논문접수 : 2015년 11월 9일
(Received 9 November 2015)

논문수정 : 2016년 1월 25일
(Revised 25 January 2016)

심사완료 : 2016년 2월 1일
(Accepted 1 February 2016)

Copyright©2016 한국정보과학회 : 개인 목적이거나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.
정보과학회 컴퓨팅의 실제 논문지 제22권 제4호(2016. 4)

showed that the algorithm could distinguish between the trajectory within 10 meters and the distant trajectory with 95% accuracy and, at worst, 65% of calculation reduction, as compared with the discrete Fréchet distance.

Keywords: vehicle GPS data, trajectory distance, Fréchet distance, similarity of trajectory

1. 서론

스마트 기기의 보급에 따라, GPS를 이용한 위치 정보 수집은 흔히 이용되는 서비스이다. 수집된 위치 정보를 순서대로 연결하면 점의 서열로 이루어진 다중곡선 궤적이 생성된다. 이렇게 수집된 궤적을 통해 전자 지도를 수정하거나 생성해내는 연구도 활발하다[1]. 이를 위해서는, 0같은 도로 상의 궤적들을 모아 궤적 군집을 생성하는 작업과 각 군집 내부 궤적들의 이동 추세를 대표하는 대표 궤적을 추출해 내는 작업이 필요하다. 이 중에서 같은 도로 상의 궤적들을 모으는 문제는 곡선 사이의 유사성 판별 문제로 볼 수 있다. 이러한 곡선 간의 비교문제는 여러 분야에서 연구가 진행되고 있다[2,3].

어떤 두 궤적이 유사한지를 측정하는 방법으로는 프레세 거리가 있다[4]. 프레세 거리는 두 곡선 사이의 거리를 진행 방향을 고려하여 나타낸 값이다. 이 값을 이용하면 같은 도로를 달리는 궤적들을 모을 수 있지만 궤적의 양이나 점의 개수가 많을 경우 시간 소모가 심하기에 새로운 방법이 필요하다. 따라서 본 연구는 기존의 이산 프레세 거리를 이용하여, 두 궤적이 같은 도로를 달리는 궤적인지를 빠르게 판별할 수 있는 방법을 제시하는데 목적을 두었다.

2. 기존 방법

궤적들 간의 거리 계산에는 흔히 하우스도르프 거리와 프레세 거리가 활용된다. 하우스도르프 거리는 두 곡선에 대해서, 한 곡선 위의 각 점에서 다른 곡선까지의 최소 거리를 모두 측정하여, 이 중에서 가장 긴 거리를 통해 두 궤적 사이의 거리를 나타낸다. 하지만 하우스도르프 거리는 두 궤적의 진행 방향이나 모양을 고려하지 않기 때문에 실제 두 궤적의 유사성을 판단하기엔 부적합하다.

프레세 거리는 하우스도르프 거리와는 다르게, 두 궤적의 진행 방향이나 모양을 고려하여 계산되기 때문에 그 값이 작을수록 두 궤적은 가깝고 모양도 비슷한 유사 궤적이라고 판단할 수 있다. 프레세 거리의 측정은 개와 사람의 산책을 예로 쉽게 설명된다. 목줄을 묶은 개와 사람이 각자 궤적을 따라 시작 지점부터 끝 지점까지 산책을 한다. 각자의 속도는 조절할 수 있으나 뒤로 돌아갈 수는 없다. 이 때 이 산책을 가능하게 하는 목줄의 최소 길이가 프레세 거리가 된다. 두 곡선이 다중선분인 경우, 각각 점의 개수가 m, n 인 두 다중선분

P, Q 에 대해 연속 프레세 거리는 $O(mn \log(mn))$ 에 계산된다[5].

이산 프레세 거리는 기존의 연속적인 탐색이 아니라, 다중선분을 이루는 각 점에서의 거리만 측정하는 방법이다. 그림 1은 연속 프레세 거리와 이산 프레세 거리의 계산 방법의 차이점을 보여준다. 이산 프레세 거리는 끈으로 묶여있는 한 쌍의 개구리를 이용하여 쉽게 설명할 수 있다. 끈으로 묶여있는 두 개구리가 각자 징검다리에 놓여있다. 두 개구리는 한 번에 한 칸 이동하거나 이동하지 않을 수 있지만 뒤로 돌아갈 수는 없다. 두 개구리의 매순간 위치의 쌍을 ‘커플링(coupling)’이라고 하고, 이때의 끈 길이를 ‘커플링의 길이’라고 한다. 끈의 길이를 최소화하면서 마지막 지점까지 이동했을 때, 가장 길었던 커플링의 길이가 이산 프레세 거리이다. 이산 프레세 거리는 각각 점 개수 m, n 개로 이루어진 두 다중선분 P, Q 에 대해 동적 프로그래밍을 통해 $O(mn)$ 에 계산이 가능하다[6]. 계산 방법은 알고리즘 1과 같다[7].

일반적인 프레세 거리들은 최대 길이 값 하나만 나타내기 때문에 궤적에 포함된 잡음의 영향을 크게 받는 문제가 있다. 이를 해결하기 위해 제시된 평균 이산 프레세 거리는 두 궤적 사이의 커플링의 길이 평균이 최소가 되도록 탐색하는 방법이다. 평균적인 길이 값을 이용하기 때문에, 두 궤적이 대부분의 구간에서 가까울 경우 일부잡음이 발생하더라도 두 궤적 사이의 평균 거리 측정에는 크게 영향을 미치지 않는다. 평균 이산 프레세 거리도 이산 프레세 거리와 같은 $O(mn)$ 에 계산이 가능하다.

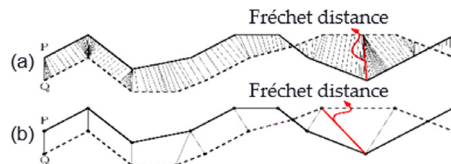


그림 1 두 가지 프레세 거리의 비교. (a) 연속 프레세 거리 (b) 이산 프레세 거리. 이산 프레세 거리는 계산 횟수가 연속 프레세 거리보다 짧지만 결과 길이가 더 길 가능성이 있다.

Fig. 1 Comparison of two concepts of Fréchet distance. (a) Continuous Fréchet distance. (b) Discrete Fréchet distance. Discrete Fréchet distance can be calculated faster than the continuous Fréchet distance although the result of the measure might be larger

Algorithm 1 Calculation of the discrete Fréchet distance

```

1: input   :  $T_P = \{p_0, p_1, \dots, p_m\}, T_Q = \{q_0, q_1, \dots, q_n\}$ 
2: output  : a fréchet distance between  $T_P$  and  $T_Q$ 
3:  $\|p_i - q_j\|$  : distance between  $p_i$  and  $q_j$ 
4: procedure DISCRETE FRÉCHET DISTANCE  $F(T_P, T_Q)$ 
5:    $D_{0,0} \div \|p_0 - q_0\|$ 
6:   for  $j \div 1$  to  $n$  do  $d_{0,j} \div \max\{d_{0,j-1}, \|p_0 - q_j\|\}$ 
7:   end for
8:   for  $i \div 1$  to  $m$  do
9:      $d_{i,0} \div \max\{d_{i-1,0}, \|p_i - q_0\|\}$ 
10:    for  $j \div 1$  to  $n$  do
11:       $d_{i,j} \div \max\{\min(d_{i-1,j-1}, d_{i-1,j}, d_{i,j-1}),$ 
12:         $\|p_i - q_j\|\}$ 
13:    end for
14:  end for
15:  Return  $d_{m,n}$ 
16: end procedure

```

이차(quadratic)인 이산 프레셰 거리의 계산 시간을 줄이기 위한 방법들도 많이 연구가 되었다. Agarwal et al.[8]은 $O(mn \log(\log n) / \log n)$ 의 방법을 제시하였으며, Avraham et al.[9]은 Driemel과 Har-peled[10]이 제시한 지름길(shortcuts)을 이용하여 프레셰 거리의 근사치를 $O((m^{2/3}n^{2/3} + m + n) \log(m + n))$ 의 시간에 계산해내었다. 이렇게 다양한 연구가 진행되었지만, 프레셰 거리를 $O(n^{2-\delta})$, (단, $\delta > 0$)에 계산하는 것은 불가능함을 Bringmann[11]이 설명하였다. 따라서 많은 궤적 쌍들의 유사성을 빠르게 판별하기 위해서는 정확한 값이 아닌 근사적인 방법을 고안해야 한다.

3. 제안 방법

본 논문에서는 두 궤적 사이의 이산 프레셰 거리를 이용하여, 일정 거리 이내의 거리가 가까운 유사 궤적들을 빠른 시간 안에 판별할 수 있는 방법을 제안한다. 이 방법에는 두 가지 작업이 포함되는데, 첫 번째는 점 간격 재조정 작업이고 두 번째는 범위 제한 탐색 작업이다.

3.1 점 간격 재조정

점 간격 재조정 작업은 궤적을 이루는 점의 샘플링 간격을 일정하게 맞추기 위해 수행한다. 먼저, 점 개수가 각각 m, n 개인 두 궤적 P, Q 와 이들의 재조정 점 간격을 식 (1), (2)와 같이 정의한다.

$$P = \langle p_i(x_i, y_i) \rangle, Q = \langle q_j(x_j, y_j) \rangle \quad (1)$$

$$\text{intv}_P = \sum_{i=0}^{m-2} \text{dist}(p_i, p_{i+1}) / (m+n) \quad (2)$$

$$\text{intv}_Q = \sum_{j=0}^{n-2} \text{dist}(q_j, q_{j+1}) / (m+n)$$

각 궤적은 2차원 점의 수열이다. $\text{dist}(x_i, x_{i+1})$ 은 두 점 x_i 와 x_{i+1} 사이의 길이이다. 재조정 점 간격은 각 궤적의 길이를 두 궤적의 점의 개수 합으로 나누어 계산한

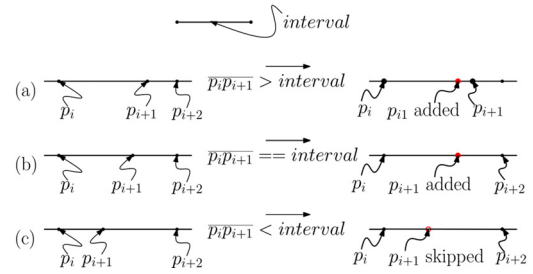


그림 2 점 간격 재조정 방법. (a)재조정 간격보다 넓은 경우 새로운 점을 추가한다. (b)재조정 간격과 같을 경우 기존의 간격을 유지한다. (c)재조정 간격보다 좁을 경우 점을 제거한다.

Fig. 2 Point reorganization method. (a)Add additional point when the distance is longer than the reorganize interval. (b)Maintain the interval when the distance is the same with the reorganize interval. (c)Skip the point when the distance is shorter than the reorganize interval

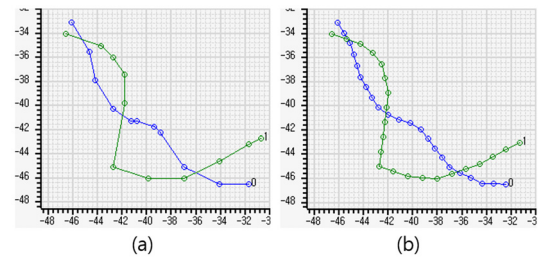


그림 3 점 간격 재조정 전 후의 비교. (a)점 간격 재조정 전, (b)점 간격 재조정 후

Fig. 3 The comparison between original trajectories and the point interval reorganized trajectories. (a) Original trajectories. (b)Point interval reorganized trajectories

다. 재조정 점 간격을 이용하여 그림 2와 같이 점 간격을 재조정한다. 궤적을 이루는 점을 순서대로 확인하면서 점 간격이 재조정 간격보다 클 경우 새로운 점을 추가하고, 같을 경우 기존의 점을 유지한다. 점 간격이 재조정 간격보다 좁을 경우 해당 점을 궤적에서 제거한다. 점 간격 재조정 결과로, 두 궤적은 간격이 일정한 최대 $m+n$ 개의 점을 갖게 된다. 그림 3은 점 간격 재조정 결과의 예를 보여준다. 점 간격 설정 작업에 $O(m), O(n)$ 이 수행되고 보간 및 생략 작업에 $O(m+n)$ 이 2회 수행된다. 두 궤적의 점의 개수가 n 으로 같다고 가정했을 때, 점 간격 재조정 작업에는 총 $O(6n)$ 만큼의 시간이 소모된다.

3.2 범위 제한 탐색

범위 제한 탐색 작업은 이산 프레셰 거리의 계산 특

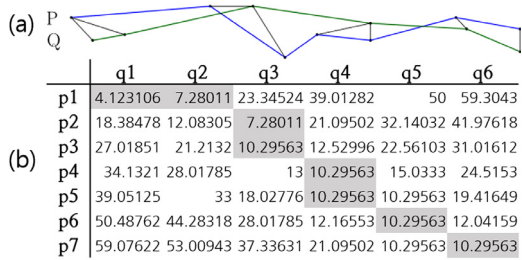


그림 4 이산 프레셰 거리 측정 결과. (a) 궤적 P와 Q 사이의 점연색 선분은 각 점에서의 커플링을 나타낸다. (b) 이산 프레셰 거리 계산 테이블. 주황색으로 표시된 값들은 커플링 결과이다.

Fig. 4 Result of discrete Fréchet distance calculation. (a) The black line between trajectory P and Q indicates the resulted coupling. (b) Dynamic programming table. The marked cells corresponding to the coupling

정을 이용한 방법이다. 이산 프레셰 거리는 동적 프로그래밍을 이용하여 계산되는데, 계산에 실제 활용되는 값들 외에는 계산을 생략하여 계산 횟수에서 이득을 얻을 수 있다. 예를 들어 그림 4(a)에서와 같이 진행 형태가 비슷한 두 궤적의 이산 프레셰 거리를 계산할 때에는 그림 4(b)에 나타난 영역만으로도 계산을 수행할 수 있다. 반면, 진행형태가 비슷하지 않다면 더 넓은 영역을 탐색해야 하므로 탐색영역이 줄어들수록 계산 결과는 증가한다.

위 특성을 이용하여, 이산 프레셰 거리의 근사치는 대각선 테이블을 기준으로 일정한 탐색 범위 내에서 최적인 값으로 계산된다. 계산 결과, 유사한 형태의 궤적은 원래 프레셰 거리의 계산 결과와 비슷한 근사치를 갖게 되며, 그렇지 않을 경우 원래 프레셰 거리보다 큰 값을 갖게 된다. 이 계산 결과 값을 두 궤적 사이의 유사성 판단 척도로 활용한다. 탐색 범위를 b 로 설정하고, 두 궤적의 점 개수가 n 으로 같다고 했을 때 범위 제한 탐색에는 $O(b \cdot 2n)$ 만큼의 시간이 걸린다.

3.3 유사성 판단

점 간격 재조정 작업과 범위 제한 탐색을 통해 평균 이산 프레셰 거리의 근사치를 계산한다. 본 연구의 목표는 같은 도로 상의 궤적들을 빠르게 판별하는 것이기 때문에, 정확한 거리 대신 근사치를 계산하여 유사성을 판별한다. 두 궤적의 이산 프레셰 거리 근사치가 제한 거리 d 이하일 때 이 두 궤적을 유사 궤적이라고 정의하고, 유사 궤적은 식 (3)과 같이 판별한다.

$$S(d, P, Q) = \begin{cases} ADFD(b, P, Q) \leq d, & T \\ ADFD(b, P, Q) > d, & F \end{cases} \quad (3)$$

$ADFD(b, P, Q)$ 는 궤적 P, Q 에 대해 탐색 범위가 b 인 평균 이산 프레셰 거리 근사치 계산 결과를 나타낸다. 두

궤적의 점의 개수가 n 으로 같을 때, 판별을 위한 시간은 간격 재조정의 $O(6 \cdot n)$ + 범위 제한 탐색의 $O(b \cdot 2n)$ 으로 $O((2b + 6)n)$ 만큼의 시간이 걸린다.

4. 실험 및 평가

기존 평균 이산 프레셰 거리를 기준으로, 제안 방법인 평균 이산 프레셰 거리 근사치의 정확도와 계산 횟수 이득에 대한 실험을 수행하였다.

4.1 실험 설정

본 실험에 사용될 궤적들의 실제 모양은 그림 5와 같다. 데이터의 설정은 표 1과 같다. 강남과 성남 일대에서 수

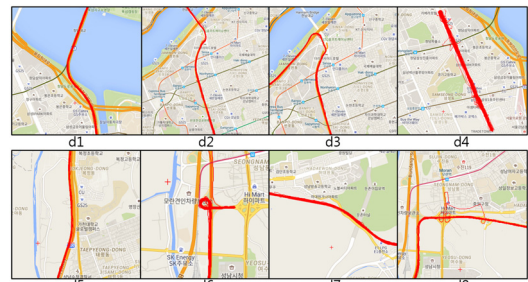


그림 5 실제 도로에서의 실험 데이터 궤적. 붉은 선은 궤적 데이터를 나타낸다.

Fig. 5 Actual trajectory on the map. The red line represents the trajectory data

표 1 실험 데이터 설정. (A) 궤적들의 점 개수 분포 범위. (B) 궤적들의 점 개수 평균

Table 1 Data setup for experiment. (A) The range of points distribution of trajectories. (B) The average number of points of trajectories

Area	Data	(A)	(B)
Gangnam	d1	233 ~ 1319	334
	d2	321 ~ 1913	597
	d3	409 ~ 2555	740
	d4	169 ~ 1048	343
Sungnam	d5	115 ~ 1637	139
	d6	82 ~ 520	116
	d7	71 ~ 192	86
	d8	172 ~ 859	228

집된 약 1주일 분량의 궤적 중에서 일부를 추출한 것으로, 각 데이터 집합은 300개의 궤적 모음이다. 궤적의 점 개수가 탐색 제한 거리에 미치는 영향을 확인하기 위해 각 집합에 대해 궤적들의 점 개수 분포 범위(A)와 그 평균(B)을 나타내었다.

4.2 유사성 판별 정확도 실험

기존 이산 프레셰 거리와의 비교를 통해 유사성 판별

표 2 탐색 범위와 정확도의 관계 실험. (A) 평균 이산 프레셰 거리가 10m 이내인 궤적 쌍 수. (B) 제안 방법으로 찾아낸 10m 이내의 궤적 쌍 수

Table 2 Experiment on the relationship between search range and accuracy. (A) Number of a pair of trajectories within 10 meter calculated by the average discrete Fréchet distance. (B) Number of a pair of trajectories within 10 meter calculated by the suggested method

(A)	Range	(B)	Accuracy (%)
44319	0	35903	81.10
	1	39808	89.82
	2	41832	94.39
	3	42733	96.42
	4	43206	97.49
	5	43474	98.10

표 3 거리와 탐색 범위 관계 실험. (A) 거리가 d 이내인 궤적 쌍의 수. (B) 정확도가 95% 이상인 최소 탐색 범위 값

Table 3 Experiment on the relationship between distance and search range. (A) Number of trajectories within d . (B) Minimum search range that makes the more than 95% of accuracy

d (m)	(A)	(B)	Accuracy (%)
10	44319	3	96.42
15	44849	0	95.95
20	44850	0	98.92
25	44850	0	99.31
30	44850	0	99.44

을 어느 정도 정확도로 찾아내는지 확인하였다. 표 2는 데이터 $d1$ 에 대해 거리를 10m로 고정해두었을 때 탐색 범위를 0부터 5까지 변경하면서 판별 정확도를 확인해 본 결과를 나타낸다. 표에 따르면, 범위가 3이상일 때부터 판별 정확도가 95%를 넘어가는 것을 확인하였다. 탐색 범위가 커질수록 정확도가 높아지는 것도 확인된다.

표 3은 정확도를 95% 이상으로 고정했을 때 인접 궤적까지의 거리와 탐색 범위 간의 관계에 대한 실험 결과이다. 데이터 $d1$ 에 대해 거리가 10m 이내인 궤적 쌍을 판별할 때는 범위를 3 이상으로 설정해야 95% 이상의 정확도를 가질 수 있었다. 찾고자 하는 인접 궤적과의 거리가 짧을수록 궤적의 잡음에 영향을 많이 받아, 테이블의 더 넓은 부분을 탐색해야 하기 때문이다.

표 4는 전체 실험 데이터에 대해서 거리 10m 기준, 정확도가 95% 이상이 되는 최소의 탐색 범위 값을 측정한 결과이다. 실험 결과, 최소 탐색 범위가 각 테스트 데이터 집합에 따라 다른데, $d1 \sim d4$ 에 필요한 최소 탐색 범위가 $d5 \sim d8$ 에 비해 상대적으로 넓은 것을 확인할 수 있

표 4 데이터 별 최소 탐색 범위 실험 결과. (A) 평균 이산 프레셰 거리 기준 10m 이내인 궤적 쌍의 수. (B) 정확도가 95% 이상이 되도록 하는 최소 탐색 범위

Table 4 Result of minimum search range experiment. (A) Number of trajectories within 10 meter calculated by the average discrete Fréchet distance. (B) Minimum search range that makes the accuracy more than 95%

data	(A)	(B)	Accuracy (%)
d1	44319	3	96.42
d2	39681	5	95.01
d3	43475	7	95.10
d4	28517	6	95.07
d5	39637	1	99.21
d6	39153	2	96.39
d7	40952	1	99.21
d8	41197	2	97.83

었다. 이는 데이터 집합을 구성하는 궤적들에 잡음이 심한 정도에 영향을 받기 때문이다. 데이터 $d1 \sim d4$ 는 강남구 도심지의 궤적 집합이라 근처 건물들로 인해 궤적에 포함된 잡음이 많다. 반면 $d5 \sim d8$ 은 성남시 고속도로 인근에서 수집된 궤적들로 인근 건물이 적고 궤적에 포함된 잡음도 상대적으로 적었기 때문에 이러한 결과가 나타났다.

4.3 계산 횟수 이득 실험

각 데이터의 계산 횟수 이득 실험을 수행한 결과는 표 5와 같다. 정확도를 95% 이상으로 유지하기 위해서, 앞의 실험에서 가장 컸던 탐색 범위인 7을 기준으로 실험을 수행하였다. 표에 따르면, $d2$, $d3$ 와 같이 궤적들의 평균적인 점 개수가 많은 데이터에서 제안 방법을 통한 계산 이득이 큰 것을 확인할 수 있다.

표 5 계산 이득 실험 결과. (A) 이산 프레셰 거리를 이용한 계산 횟수, (B) 제안하는 이산 프레셰 거리 근사치를 이용한 계산 횟수를 나타낸다.

Table 5 Result of calculation reduction experiment. (A) Number of calculation of the discrete Fréchet distance. (B) Number of calculation of the suggested method

data	(A)	(B)	gain (%)
d1	4,998,749,523	44,564,440	99.11
d2	15,933,960,667	797,624,369	94.99
d3	24,527,928,987	989,587,830	95.97
d4	5,267,921,366	455,189,367	91.36
d5	858,432,075	183,787,044	78.59
d6	595,952,497	152,567,503	74.40
d7	331,142,820	113,103,203	65.84
d8	2,330,127,486	304,141,034	86.95

5. 결론 및 추후연구

본 논문에서는 같은 도로 위를 달리는 유사 궤적들을 빠르게 판별할 수 있는 방법에 대해 제안하고 그 정확도와 계산 이득을 실험으로 확인하였다. 이 방법은 평균 이산 프레스세 거리의 근사치를 계산하여 어떤 궤적 쌍이 찾고자하는 거리 내에 있는지 빠르게 판별한다. 본 연구의 결론은 다음과 같다.

- 이산 프레스세 거리 계산 테이블 전체를 탐색하지 않고 일부만을 탐색했을 때도 두 궤적의 거리 유사성 계산이 가능함을 확인하였다.
- 평균 이산 프레스세 거리 근사치를 계산하여, 같은 도로 위의 궤적들을 빠르게 판별하는 방법을 제안하였다. 점 개수가 m, n 인 두 궤적 P, Q 가 주어졌을 때, 기존의 평균 이산 프레스세 거리는 $O(mn)$ 에 계산할 수 있었으나, 제안 방법을 이용하면, 탐색 범위가 b 일 때 $O((b+3)(m+n))$ 에 평균 이산 프레스세 거리의 근사치를 계산할 수 있다.
- 실험 결과, 탐색 범위를 7로 설정했을 때 제안 방법은 기존의 이산 프레스세 거리대비 95%의 정확도로 인접 궤적의 탐색이 가능하며, 최소 65% 만큼의 계산 이득을 얻을 수 있었다.

현재 잡음이 가장 중요한 문제이기 때문에, 추후 연구에서는 잡음을 다루고자 한다. 4.2의 표 4에서 확인했던 바와 같이, 궤적 내부에 잡음이 심할 경우 필요한 최소 탐색 범위가 크게 증가하게 된다. 도심과 교외의 궤적 데이터를 구분할 수 없기 때문에, 잡음이 심한 궤적들에 대한 범위 변화 및 정확도의 측정 연구가 필요하다.

References

- [1] C. Lili and K. John, "From GPS traces to a routable road map," *Proc. of 17th ACM SIGSPATIAL*, pp. 3-12, 2009.
- [2] W. Rodriguez, et al., "3-dimensional curve similarity using string matching," *J. of Robotics and Autonomous Systems*, Vol. 49, 3, pp. 165-172, 2004.
- [3] B. Liao, Y. Zhang, K. Ding, and T.-m. Wang, "Analysis of similarity/dissimilarity of dna sequences based on a condensed curve representation," *J. of Molecular Structure: THEOCHEM*, Vol. 717, pp. 199-203, 2005.
- [4] A. Efrat, Q. Fan, and S. Venkatasubramanian, "Curve matching, time warping, and light fields: New algorithms for computing similarity between curves," *J. of Mathematical Imaging and Vision*, Vol. 27, pp. 203-216, 2007.
- [5] Alt H, Godau M., "Computing the Fréchet distance between two polygonal curves," *International Journal of Computational Geometry & Applications*, Vol. 5, 01n02, pp. 75-91, 1995.
- [6] Eiter T., Mannila H., "Computing discrete Fréchet distance," Citeseer, 1994.
- [7] A. Mosig and M. Clausen, "Approximately matching polygonal curves with respect to the Fréchet distance," *J. of Computational Geometry*, Vol. 30, pp. 113-127, 2005.
- [8] P. K. Agarwal, R. B. Avraham, H. Kaplan and M. Sharir, "Computing the Discrete Fréchet Distance in Subquadratic Time," *SIAM Journal on Computing*, Vol. 43, No. 2, pp. 429-449, 2014.
- [9] R. B. Avraham, O. Filtser, H. Kaplan and M. J. Katz, "The Discrete and Semi-continuous Fréchet Distance with Shortcuts via Approximate Distance Counting and Selection," *Proc. of 30th SOCG*, pp. 377-386, 2014.
- [10] A. Driemel and S. Har-Peled, "Jaywalking your Dog - Computing the Fréchet Distance with Shortcuts," *Proc. of 23th ACM-SIAM Symp. on Discrete Algorithms*, pp. 318-337, 2012.
- [11] K. Bringmann, "Why walking the dog takes time: Fréchet distance has no strongly subquadratic algorithms unless SETH fails," *FOCS, 2014 IEEE 55th Annual Symposium on*, pp. 661-670, 2014.



박진관

2015년 부산대학교 정보컴퓨터공학부 졸업(학사). 2015년~현재 부산대학교 전기전자컴퓨터공학과 석사과정. 관심분야는 알고리즘, 데이터공학, 데이터시각화 등



김태용

2015년 부산외국어대학교 임베디드IT과 졸업(학사). 2015년~현재 부산대학교 전기전자컴퓨터공학과 석사과정. 관심분야는 알고리즘, 지리정보시스템, 데이터시각화 등



박보국

2014년 부산대학교 정보컴퓨터공학과 졸업(학사). 2014년~현재 부산대학교 전기전자컴퓨터공학과 석사과정. 관심분야는 데이터마ining, 데이터시각화 등



조환규

1984년 서울대학교 계산통계학과(이학사) 1986년 KAIST 대학원 전산학과(공학석사). 1990년 KAIST 대학원 전산학과(공학박사). 1990년~현재 부산대학교 전기컴퓨터공학부 교수. 관심분야는 알고리즘, 응용그래프이론, 생물정보학