

수집된 경로데이터를 사용하는 내비게이션을 위한 대용량 경로조합 방법

구광민[†], 이태호^{**}, 박희민^{***}

A Big-Data Trajectory Combination Method for Navigations using Collected Trajectory Data

Kwang Min Koo[†], Taeho Lee^{**}, Heemin Park^{***}

ABSTRACT

In trajectory-based navigation systems, a huge amount of trajectory data is needed for efficient route explorations. However, it would be very hard to collect trajectories from all the possible start and destination combinations. To provide a practical solution to this problem, we suggest a method combining collected GPS trajectories data into additional generated trajectories with new start and destination combinations without road information. We present a trajectory combination algorithm and its implementation with Scala programming language on Spark platform for big data processing. The experimental results proved that the proposed method can effectively populate the collected trajectories into valid trajectory paths more than three hundred times.

Key words: Trajectory-based Navigation System, Location-Based Services, Combining GPS Trajectories, Big Data Processing

1. 서 론

현대 교통관련 서비스에서 큰 역할을 하는 분야 중 하나가 내비게이션이다. 내비게이션 시스템의 가장 중요한 기능은 출발지에서 원하는 목적지까지 최단 시간에 도착할 수 있는 경로의 탐색이다. 따라서, 단순히 경로만을 찾아주는 것이 아닌 가장 빠르고 교통량이 적은 최적의 경로를 찾는 문제에 대한 많은 관심과 연구가 진행되고 있다. 이를 위해 많은 연구와 방법들이 시도되고 개발되어지고 있다. 일반적으로

로 수집한 교통량 데이터의 패턴을 분석하거나 실시간 교통량 데이터를 반영하는 등의 방법이 적용되고 있다[1]. 또한, 도로상의 실제적인 상황과 운전자의 축적된 경험을 활용하기 위해 실제 운행하며 수집한 경로데이터를 기반으로 경로를 탐색하는 내비게이션 시스템도 제안되었다[2]. 이 시스템에서는 경로 안내를 할 때 수집된 경로데이터를 직접 사용하기 때문에 데이터의 양이 곧 안내가 가능한 경로의 수로 직결된다. 또한, 수집된 경로데이터들을 비교하여 사용자 요청에 비추어 최적의 경로를 찾는 방식이므로

* Corresponding Author : Heemin Park, Address: #C303, 31 Sangmyungdae-gil, Dongnam-gu, Cheonan, TEL: +82-41-550-5367, FAX: +82-41-550-5369, E-mail: heemin@smu.ac.kr

Receipt date : Jan. 22, 2016, Approval date : Jan. 28, 2016

[†] Department of Computer Software Engineering, Sangmyung University
(E-mail: codingple@gmail.com)

^{**} Department of Computer Software Engineering, Sangmyung University
(E-mail: developdestroyer@gmail.com)

^{***} Department of Computer Software Engineering, Sangmyung University

* This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (NRF-2013R1A1A1076060).

다양하고 방대한 데이터가 있을수록 더 좋은 경로를 찾을 수 있다. 즉, [2]과 같은 수집된 경로데이터를 사용하는 내비게이션은 방대한 양의 다양한 경로데이터를 필수적으로 필요하게 된다. 그런데, 방대한 양의 모든 가능한 출발-도착 위치 조합에 대한 경로데이터를 직접 수집하는 것은 매우 어렵고 한계가 있다. 또한, 특정 출발점과 도착점 사이에도 수많은 대안 경로들이 존재하기 때문에 모든 가능한 경로에 대해 각각 별도의 데이터를 수집하는 것은 불가능하게 된다. 이러한 문제점을 해결하기 위해 본 논문에서는 수집된 일부의 대용량 경로데이터를 조합하여 기존에 수집되지 않은 새로운 경로데이터로 생성하는 방법을 제안한다.

본 논문에서 제안하는 알고리즘은 도로정보를 사용하지 않으며, GPS 위치정보의 집합으로 이루어진 경로데이터만 다룬다. 수집된 경로데이터를 두 경로씩 서로 비교하고, 같은 교차로를 사용하는 지점을 기준으로 경로를 나눈 후, 나누어진 경로 중 서로 다른 부분을 교환하여 조합하는 방식이다. 조합하기 전, 경로를 조합할 가치가 있는지 확인을 한 후 조합한다. 예를 들어, 기존 경로데이터에 존재하지 않는 다른 방향으로 진행하는 경로로 조합될 때 새로운 경로로 생성하게 된다. 제안하는 알고리즘은 GPS 위치정보로 구성된 대용량 경로데이터를 처리해야 하기 때문에 경로조합 알고리즘은 스칼라 프로그래밍 언어[3]를 이용해 스파크[4] 빅데이터 처리 플랫폼 상에서 구현하였다.

2. 관련 연구

최근 들어 대용량 교통 및 경로데이터에 기반한 위치기반서비스 등에 대한 연구가 주목받기 시작했다[1, 5, 6, 7]. 예를 들어, 과거 교통 정보를 분석하여 각 경로들의 교통상황을 예측하고 버스의 지연 시간을 계산해서 최적 경로를 안내해 주는 연구가 있다[5]. 하지만, 실제 시스템 구현에 대한 구체적 설명이 없으며 경로 수집 방법이나 빅데이터 경로의 처리 방법에 대해서는 구현되어 있지 않은 문제점이 있다. 또한, 하둡과 맵리듀스를 이용해서 빅데이터 영상처리를 하고 이를 통해 차량 이동 경로를 추적하는 시스템을 제안하였다[6]. 빅데이터를 이용해서 차량 경로를 추적하는 기능은 의미가 있지만 목적지까지의

최선의 경로를 추천해주지는 않는 문제점이 있다.

수집한 경로데이터를 직접 사용하여 경로안내를 하는 내비게이션은 일반적인 내비게이션 시스템에서 데이터를 사용하는 방식과 다르다. 기존 내비게이션 시스템에서는 수집한 데이터를 수학적 계산으로 패턴을 분석하여 앞으로의 교통량을 예측하는 방식이 사용되었다[1]. 경로데이터를 이용한 경로 안내는 '경험적 지식'을 사용할 수 있게 한다. 경험적 지식은 예를 들어, 한 지역에서 오랜 기간 동안 거주한 사람이 그 지역교통에 대한 경험을 바탕으로 운전하여 최적의 경로를 찾는 것을 말한다. 지역에 대한 경험적 지식을 바탕으로 축적된 경로 중 사용자가 원하는 성향의 경로를 추출하는 방식이다. 경험적 지식은 패턴 분석을 위해 일반화된 정보가 아닌, 수집한 데이터 자체에서만 사용할 수 있다. 따라서, 경험적 지식 데이터를 사용한 경로데이터 기반 내비게이션은 기존 패턴분석 방식에서 찾을 수 없는 최적의 경로를 찾을 수 있다[7, 8]. 실제 경로를 수집하기 때문에 데이터량은 대용량화 된다. 실시간 서비스를 위해서는 빠른 경로 검색이 필요하게 되며 이를 위해 빅데이터 처리 기법을 이용해 경로를 유형별로 분류해 놓는 색인화 작업을 선처리(pre-processing)하는 방식이 이용된다.

수집 경로데이터를 사용하여 경로안내를 하는 내비게이션 시스템은 Fig. 1과 같은 시스템을 사용한다. 이러한 경로데이터 기반 내비게이션은 많은 경로데이터가 필요하게 된다. 수집된 데이터가 없는 경로에 대해서는 안내를 하는 것이 불가능하기 때문이다. 그리고 다양한 데이터가 있어야 더 좋은 경로를 찾을 수 있으며, 비교적 최적의 경로임을 알 수 있다. 하지만 앞서 언급했듯이, 다양한 경로를 수집하기에는 한계가 있다. 길에는 수많은 교차로가 있으며 이는 다양한 방법으로 진행할 수 있음을 의미한다. 경로데이터 기반 내비게이션 시스템에서 다양한 경로를 사용하고 싶으면, 그만큼 직접 수집해야한다. 기존의 관련 연구 문헌을 면밀히 조사한 바로는 GPS 경로데이터를 조합하거나 수집된 경로데이터를 이용해 새로운 경로를 조합해 내는 연구는 시도된 적이 없는 것으로 파악되었다. 본 논문에서는 경로데이터 수집의 제약을 극복하는 방안으로 경로조합 알고리즘을 고안하여 제안한다.

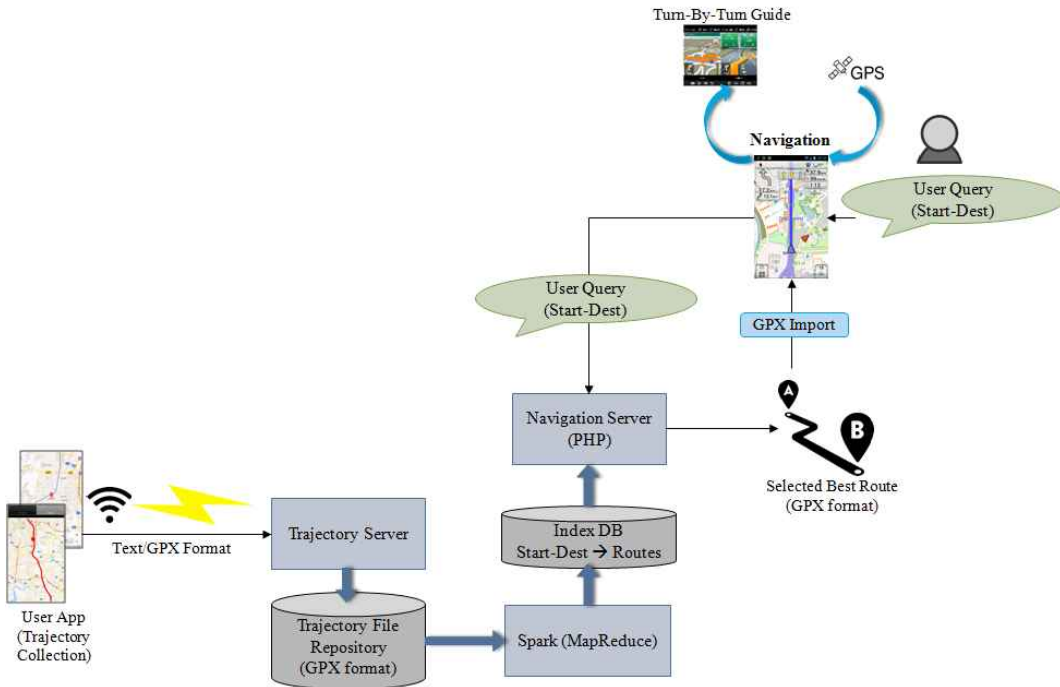


Fig. 1. Trajectory-based Navigation System Architecture.

3. 경로 조합 알고리즘

경로데이터는 GPS 위치정보들로 구성되며 위치 정보의 수가 n 이라면 경로데이터 T_i 는 $T_i = (t_i^0, t_i^1, t_i^2, \dots, t_i^{n-1})$ 와 같이 정의된다. 여기서 t_i^j 는 GPS 위치정보이다. 경로 조합은 모든 두 경로에 대해 각 경로마다 서로 한 번의 조합절차를 거친다. 두 경로를 기준으로 조합절차는 ‘거리측정 → 방향성 검사 → 조합 여부 판단 → 조합’의 순서로 진행된다. 먼저 경로 내

위치정보의 순서대로 두 경로의 GPS 지점간의 거리를 측정한다. 각 경로의 GPS 지점 하나씩, 한 쌍의 GPS 지점이 같은 교차로에 있다 판단할 수 있는 가까운 거리일 때, 이 GPS 지점들을 중심으로 두 경로의 진행 방향을 검사한다. 두 경로의 진행 방향이 판단되면 방향성을 기반으로 조합의 가치가 있는지 판단하고, 가치가 있으면 조합한다. 경로 조합 알고리즘은 Fig. 2와 같은 순서도로 표현될 수 있다.

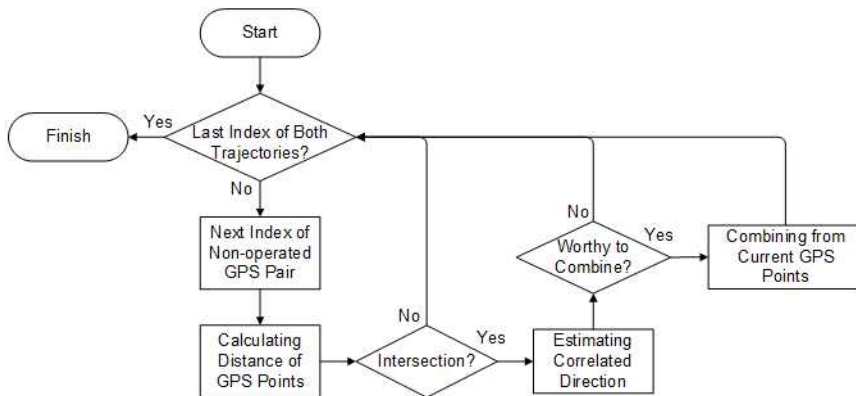


Fig. 2. Flowchart of Combining Two Trajectories.

경로조합은 기본적으로 GPS 지점과 다른 경로의 GPS 지점 사이의 거리를 이용한 상대적 위치관계의 성질을 이용한다. 이는 도로 정보를 사용하지 않고 두 경로데이터의 관계를 얻기 위한 수단이다. 같은 교차로를 이용했다고 가정할 만큼 두 경로가 가까워지면, 그 GPS 지점들을 기준으로 두 경로의 거리 변화를 측정하면 진행 방향을 판단할 수 있다. 예를 들어, 두 경로의 GPS 지점들의 거리가 멀어지면 두 경로는 서로 다른 방향으로 진행한다고 판단이 가능하다. 방향성을 판단하는 방법과 경로조합의 가치는 아래에서 설명한다.

4. 경로의 진행 방향 판단

4.1 경로 진행 방향 판단의 전제조건

경로 진행 방향 판단을 위해 전제되는 조건들이 있다. 진행 방향을 GPS 지점 간의 거리로 판단하기 때문에 생기는 조건들로, 다음과 같다.

- 1) 경로데이터는 순서(방향)가 있는 GPS 지점들의 집합이어야 한다.
- 2) GPS 지점들은 경로를 식별하기에 충분히 가까운 간격을 유지해야한다.
- 3) GPS 지점들의 위치정보 오류는 경로를 판단하기에 문제없는 수준이어야 한다.

먼저 경로데이터는 GPS 위치정보로 이루어져있으며 그 순서가 있는 리스트이다. GPS 지점들의 순서는 경로의 진행 방향을 결정하며 이를 바탕으로 두 경로의 진행 방향을 비교 판단하게 된다. 그리고 GPS 지점의 위치들이 경로의 모양을 묘사할 수 있는 간격이어야 한다. 즉, 경로데이터를 수집할 때 GPS 정보 수집시간 간격이 적절하게 설정되어야 한다. 수집시간 간격이 너무 길면 경로의 방향이나 모양에 오류가 있을 수 있고, 너무 짧으면 불필요하게 처리해야 할 데이터의 양만 늘어날 수 있다. 마지막으로 경로데이터에 큰 오류가 있는 GPS 위치정보가 없어야 한다. 예를 들어, 실내 환경이거나 차량이 운행되지 않을 때, 긴 시간 정차 시 등등 경로데이터에 정확도가 낮은 GPS 위치정보의 오류가 심한 데이터를 말한다. 경로의 진행 방향 판단은 GPS 지점 사이의 거리를 이용한 방법이기 때문에 위치정보의 정확성이 중요하다.

위 조건들이 충족되었다면 교차로를 찾기 위해 두 경로의 GPS 지점들을 차례대로 비교한다. Fig. 3과 같이 한 경로를 기준으로, 기준이 된 경로의 첫 번째 GPS 지점부터 상대 경로의 모든 GPS 지점들과 차례대로 거리를 측정한다. 기준 경로의 첫 번째 GPS 지점과 거리측정이 끝나면, 두 번째 GPS 지점을 상대 경로의 GPS 지점들과 차례대로 거리를 측정한다. 기준경로의 마지막 GPS 지점과 상대 경로의 마지막 GPS 지점의 거리측정이 끝나면, 두 경로의 진행 방향 판단을 위한 거리 측정이 끝난다. 두 경로의 진행 방향은 '같은방향', '반대방향', '다른방향' 세 가지로 판단한다.

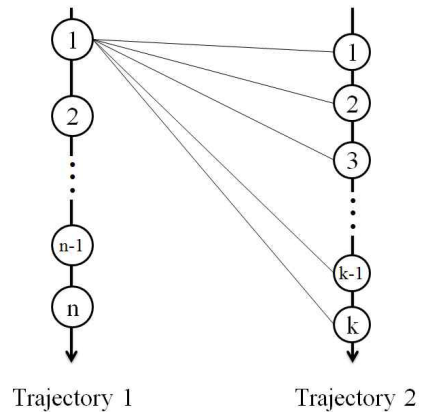


Fig. 3. Step of Calculating Distance to Estimate Inter-sections.

4.2 '같은방향'과 '반대방향'

거리측정 중, 두 GPS 지점들이 같은 교차로라고 판단되는 거리 이내면 이 점들을 중심으로 진행 방향을 판단한다. 방향은 기준이 된 두 GPS 지점들로부터 경로상의 순서를 옮겨가며 거리를 측정하고, 증가와 감소의 패턴을 나열한다. 먼저 같은 방향인지 확인하는 방법은 다음과 같이 정의된다.

$$d_i = \frac{1}{T_1^{(1+i)/2}} \frac{1}{T_2^{(2+i)/2}} \quad (i = 1, 2, \dots, N) \quad (1)$$

$$g(j) = d_{j+1} - d_j \quad (j = 1, 2, \dots, N-1) \quad (2)$$

$$D(j) = \begin{cases} \text{increase} & g(j) \geq 0 \\ \text{decrease} & g(j) < 0 \end{cases} \quad (3)$$

식 (1)에서 T_1 과 T_2 는 비교하는 두 경로이며, 기준 경로에 영향을 받지 않는다. T_1^1 과 T_2^1 은 각 경로에 속하는 기준이 된 GPS 지점들이며, N은 미리 정해진

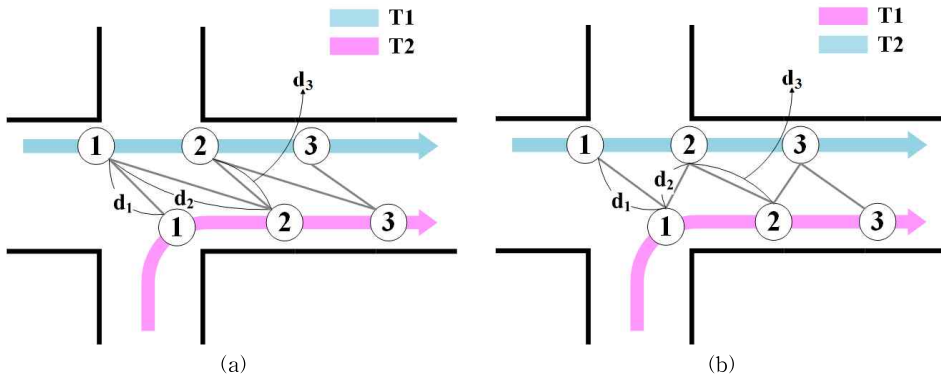


Fig. 4. Example for Patterns of 'Same-Direction'. (a) *increase-first* Pattern and (b) *decrease-first* Pattern of Same Direction.

거리측정 횟수이다. 그리고 식 (3)에서 $D(\cdot)$ 는 $N-1$ 개의 증가(increase) 혹은 감소(decrease) 값을 가지는 리스트이다. 이 때 $D(\cdot)$ 가 증가와 감소가 진동하는 패턴일 때 '같은방향'으로 판단한다.

Fig. 4의 (a)는 '같은방향'으로 판단하는 경우 중 증가로 시작하는 패턴이다. T_1^1 과 T_2^1 은 같은 교차로에 있으므로 근접한 위치에 있다. 그러므로 d_1 에 비해 d_2 는 증가한다. 이 때 만약 T_1 과 T_2 가 같은 진행 방향이면, T_2^1 가 T_1^1 에 비해 T_2^2 와 가까운 위치에 있을 것이다. 따라서 d_2 에 비해 d_3 는 감소한다. 이를 반복하면, $D(\cdot)$ 는 증가와 감소의 진동패턴이 된다. 반면, (b)는 감소로 시작한다. T_1^1 이 T_2^1 과 근접한 위치임에도 불구하고 T_2^2 와 더 가까울 수 있다. 이 경우에는 d_1 에 비해 d_2 가 감소하지만, 제안한 방법으로 판단하면 역시 증가와 감소의 진동패턴이 된다.

'반대방향'에 대한 판단은 '같은방향' 검사와 방법은 같지만, 검사할 경로의 반대방향으로 GPS 지점 검사 순서를 옮겨야 한다. T_2 가 T_1 의 '반대방향'인지 확인할 때 다음과 같이 정의한다.

$$d_i = \frac{T_1^{\lfloor (4-i)/2 \rfloor} T_2^{\lfloor (2+i)/2 \rfloor}}{T_1^{\lfloor (4-i)/2 \rfloor} T_2^{\lfloor (2+i)/2 \rfloor}} \quad (i = 1, 2, \dots, N) \quad (4)$$

단, T_1^1 을 중심으로 이전 GPS 지점들의 순서는 ' $T_1^0, T_1^{-1}, T_1^{-2}, \dots$ '와 같다. 이외에 식(2)와 식(3)은 똑같이 적용한다.

증가와 감소가 진동패턴을 보이는지 확인하기 위해서는 적어도 $N \geq 4$ 이어야 한다. N 이 너무 작으면 정확성이 떨어지고, 너무 크면 다른 교차로까지 진행해버릴 수 있기 때문에 GPS 지점들의 간격을 고려하여 적절하게 결정해야 한다.

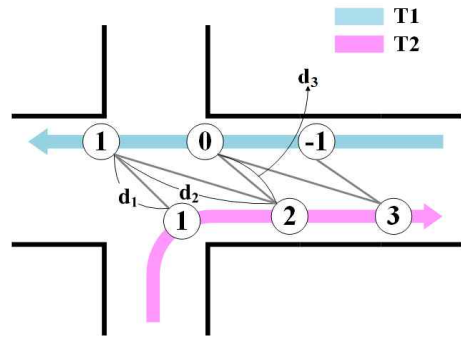


Fig. 5. A Case of Reverse Direction.

4.3 '다른방향'

먼저 방향성 검사에서 '같은방향'이나 '반대방향'이 아니라고 판정된 경우를 모두 '다른방향'으로 판단한다. 그렇지만 '다른방향'을 검출할 수 있는 특징도 있다.

Fig. 6처럼 다른 방향일 경우 두 경로사이의 GPS

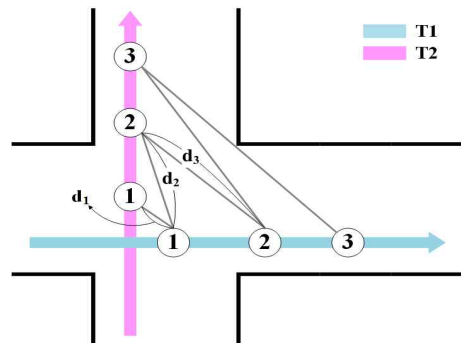


Fig. 6. A Case of Different Direction.

지점들 간의 거리는 점차 증가한다. 즉, 두 경로의 GPS 지점들 사이의 거리가 증가하므로, $D(\cdot)$ 가 연속적인 증가패턴을 갖는다. 하지만 T_1 과 T_2 의 위치, GPS 측정 오류 값으로 인해 $D(1)$ 부터 $D(N-1)$ 까지 모두 증가패턴을 갖지 않을 수도 있다. 이러한 점을 고려하여 '같은방향'이나 '반대방향'이 아닌 경우를 모두 '다른방향'으로 판단하였다.

5. 경로조합의 가치

경로의 진행 방향이 판단되면 경로조합을 했을 때 가치가 있는지 확인한다. 경로조합의 가치를 판단하고 가치가 있는 경우에만 경로조합을 수행한다. 경로조합의 가치는 두 경로의 상대적 방향성을 고려하여, 생성되는 경로가 유용한 경로일 때 가치가 있는 조합으로 정의한다. 유용한 경로는 조합된 경로가 기존 경로와 다른 새로운 경로여야하며, 이는 교차로에서 기존 경로와는 다른 방향으로 진행되는 경로가 될 경우이다.

경로조합의 가치판단은 경로 T_1 과 T_2 가 교차로 한 군데서 만나는 경우 두 경로를 조합한다고 할 때, (T_1 출발점 → 교차로 → T_2 도착점)과 (T_2 출발점 → 교차로 → T_1 도착점) 두 가지 경로조합에 대해 가치판단을 하게 된다.

5.1 검사 항목

Fig. 7은 T_1 을 중심으로 경로조합의 가치가 있는 가능한 경로들(T_2)이다. 먼저 두 경로가 같은 방향으로 진행되는 경우에는 조합할 필요가 없다. 그리고 T_1 의 반대방향으로 T_2 가 진행할 경우, T_1 을 중심으로 생성된 경로는 되돌아가는 경로로 생성된다. 이는 비

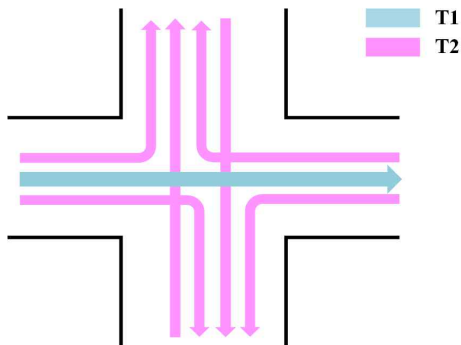


Fig. 7. T_1 's Worthy Directions to Combine.

효율적인 경로가 될 가능성이 있으며, U턴이 불가능한 교차로에서 잘못된 경로안내를 할 가능성이 있다. 따라서 T_1 을 중심으로한 조합은 가치가 없다. 하지만 T_2 를 중심으로 조합하면 가치가 있을 수 있으므로, T_2 는 별도로 검사해야한다. 이외의 '다른방향'들은 새로운 경로를 생성할 수 있으므로 가치가 있는 것으로 판단한다. 이를 토대로 경로조합가치를 검사할 항목은 세 가지로 정의한다.

- 1) T_1 의 '반대방향'으로 T_2 가 진행
- 2) T_2 의 '반대방향'으로 T_1 이 진행
- 3) T_1 과 T_2 가 '같은방향'으로 진행

1)은 T_1 의 조합가치를, 2)는 T_2 의 조합가치를 판단하기 위한 항목이다. 예를 들어 1)이 참일 경우, T_1 을 중심으로 T_2 와 조합할 가치가 없는 것으로 판단한다. 그리고 3)이 참일 경우, 두 경로 모두 조합가치가 없다.

5.2 판단 결과

검사한 항목들의 결과는 (T_1 출발점 → 교차로 → T_2 도착점)과 (T_2 출발점 → 교차로 → T_1 도착점)의 두 가지 경로조합에 대해 '두 경로 모두 조합', '한 경로만 조합', '조합하지 않음'까지 세 가지로 나온다. 두 경로를 모두 조합하는 경우는 검사항목 1), 2), 3)이 모두 거짓일 경우이다. 서로의 '반대방향'으로 진행하지 않으며 '같은방향'으로도 진행하지 않으면 두 경로 모두 조합에 가치가 있다. 즉 1), 2), 3)이 모두 거짓인 경우에는 새로 생성되는 경로가 두 개다. 한 경로만 조합하는 경우는 1)과 2)가 참과 거짓 하나씩 있는 경우다. 이때는 되돌아가는 경로를 생성하지 않으므로, 새로운 경로가 한 개이다. 이외에 나머지 경우는 모두 경로를 생성하지 않는다. Fig. 8에 경로조합 판단 결과를 도식화하여 보인다.

6. 경로조합 후 처리

방향성 판단과 조합가치 판단 후 경로조합이 끝나면 경로의 다음 순서에 있는 GPS 지점들과 거리를 측정한다. 이 때 같은 교차로로 판단할 거리설정에 따라, 이미 경로조합이 이루어진 교차로에서 다시 경로조합을 할 가능성이 있다. 이런 경우 불필요한 시

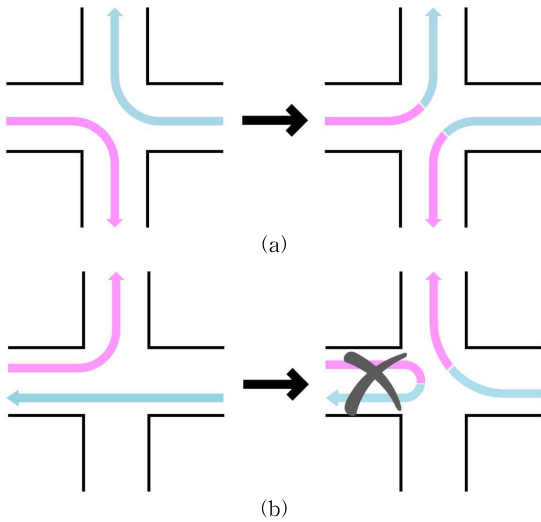


Fig. 8. Combination of (a)Both Trajectories and (b) One Trajectory.

간 낭비와 중복된 경로를 생성한다. 이러한 문제는 도로정보를 사용하지 않으며 GPS 지점 간의 거리를 이용하기 때문이다. 경로조합 후 처리는 이를 해결하기 위해 교차로로 판단되는 거리를 이탈하는 방법이다. 이 방법은 기준경로의 영향을 받는다.

Fig. 9은 경로조합 후 처리하는 방법의 기본적인 흐름을 정의한 것이다. T_1 은 기준경로이고 T_2 는 T_1 과

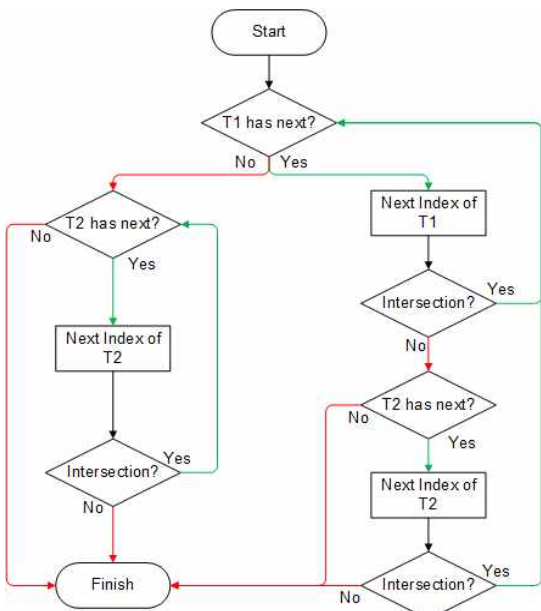


Fig. 9. Flowchart of post-process.

조합하는 다른 경로이다. 기준경로의 GPS 지점의 순서를 조금만 옮겨도 많은 연산을 줄일 수 있기 때문에 T_1 을 중심으로 진행된다. 교차로가 아니라고 판단하는 위치 또는 더 이상 거리를 비교할 GPS 지점이 없으면 종료한다. 비교할 GPS 지점이 있으면 이어서 거리측정을 한다.

위 내용은 두 경로가 조합을 하는 경우에 사용하는 경우이다. 이 방법은 불필요한 연산을 줄이기 위해 다음 두 가지 경우에도 사용한다. 두 경로가 같은 방향으로 진행할 때와 T_1 이 T_2 의 '반대방향'으로 진행할 때이다.

Fig. 10에서 P_1 은 같은 교차로에 있다고 판단된 GPS 지점들의 쌍이다. P_1 에서 두 경로는 같은 방향이라는 결과가 나온다. 이 때 위의 방식대로 두 경로의 GPS 순서를 건너뛴다. 그리고 두 경로가 마지막으로 같은 교차로의 거리로 측정된 GPS 쌍이 P_2 이다. P_2 에서 두 경로를 조합하고 다음 거리측정을 하는 방식이다. T_1 이 T_2 의 '반대방향'으로 진행할 때와 같은 방식이며, GPS 순서를 옮기는 방향을 T_2 는 반대방향으로 한다.

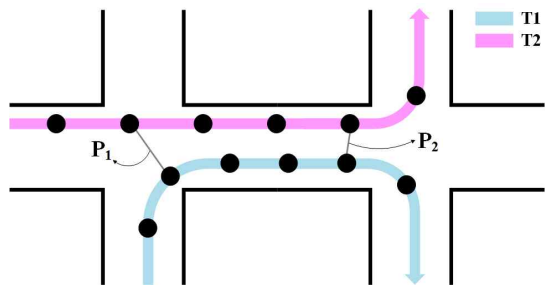


Fig. 10. A Case of Using Post-process in Same Direction.

7. 실험 결과

7.1 실험 환경

실험을 위한 경로데이터는 차량을 운행하며 직접 수집하였으며 Table 1에 수집한 경로데이터에 대한 설명을 보인다. 경로데이터는 GPS의 XML포맷인 GPX[9]이며 참여형 오픈소스 지도인 오픈스트리트맵[10]을 기반으로 한 개발된 OsmAnd[11]을 사용했다. 아래 Fig. 11은 수집한 경로들을 오픈스트리트맵 기반 맵데이터인 uMap[12]을 통해 가시화한 것이다.

제안한 알고리즘은 빅데이터 처리 플랫폼인 스카

Table 1. Collected Trajectory Data for Experiment

Collected Trajectory Data Information	
Number of Trajectories	56
File Size	13.9 MB
Area	from Seoul to Jeonju (Korea)
Used Navigation Systems	T-map, Kinggisa, iphone



Fig. 11. Visualization of Collected Trajectories.

크[4, 13]를 사용하기 위해 함수형 프로그래밍 언어인 스칼라[3, 14]를 사용하여 구현하였다. 스파크 클러스터는 3대의 슬레이브 노드와 1대의 마스터 노드, 총 4대의 서버로 설정되었으며 서버의 사양 및 스파크 버전은 다음 Table 2와 같다.

7.2 실험결과

Table 3과 같이 실험 결과 총 생성된 경로파일 수 19,168개, 파일 용량 7.11 기가바이트, 수행시간 13분 39초이다. 당초 수집된 경로 수 56개 대비 약 300배 이상의 경로가 생성되었음을 알 수 있다. 아래 Fig.

Table 2. Experimental Environments

Specification of Spark Machine	
Number of Computers	4
Main Memory	8 GB
Processor	Intel Core i7-47703.40 GHz
OS	Linux Ubuntu 14.04.2 LTS (64bit)
Hard Disk	2 TB
Spark Version	1.2.1
Scala Version	2.10.4
Java Version	1.7.0_75

Table 3. Experimental Results

Output Information	
Number of Total Trajectories	19,168
File Size	7.11 GB
Run Time	13 min 39 s

11.은 각각 경로조합 전과 후에 대한 가시화 예제이며, 특정 출발점 갖는 경로데이터들을 오픈스트리트 맵 기반 맵 에디터인 JOSM[15]을 이용해 가시화했다. Fig. 12에서 볼 수 있듯이 경로 조합을 통해 한군데의 출발점에서 시작 가능한 경로의 수가 매우 늘어났음을 알 수 있다. 이는 경로조합을 통해 내비게이

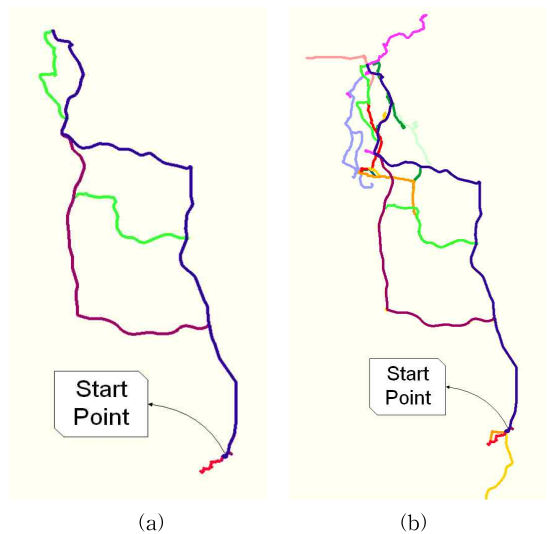


Fig. 12. An Example of Trajectory Combinations. (a) Before and (b) After Combining Trajectories.

선 시스템은 경로안내를 할 수 있는 경로의 수가 증가했으며 또한 목적지까지 더 다양한 방법의 경로를 생성함으로써, 전보다 더 최적의 경로를 찾는 것이 가능하게 되었다. 이를 통해 경로조합이 경로데이터 기반 내비게이션의 성능을 향상시킬 수 있음을 알 수 있다.

8. 결 론

본 논문에서는 도로상의 실제적인 상황과 운전자의 축적된 경험을 활용할 수 있는 경로데이터 기반으로 내비게이션 시스템을 위한 경로조합 방법에 대해 제안하였다. 도로정보에 의존하지 않고 수집된 경로데이터의 GPS 위치정보만을 이용해 경로를 조합하는 방법이다. 출발-도착점이 다른 두 경로가 교차로에서 교차할 경우 경로의 진행방향을 판단하고 조합할 가치가 있는지 판단하는 알고리즘을 고안하였다. 경로데이터는 특성상 빅데이터화되고 경로내 GPS 지점의 수가 많으므로 빅데이터 처리 플랫폼 상에서 알고리즘을 구현하였다. 함수형 언어인 스칼라를 이용해 스파크 상에서 구현하였다. 직접 수집한 경로데이터를 이용해 실험한 결과 약 300배 이상의 개수의 경로를 조합해 내는 것이 가능함을 보였다. 이를 통해 내비게이션 시스템이 경로안내를 할 수 있는 경로의 수를 증가시킬 수 있었다. 따라서, 목적지까지의 더 다양한 방법의 경로를 생성함으로써 더 최적의 경로를 찾는 것이 가능하다. 이를 통해 경로조합이 경로데이터 기반 내비게이션의 성능을 향상시키는 것을 보였다. 향후 연구로는 GPS 오류값에 강인한 알고리즘 개발 및 보다 많은 경로데이터 수집을 통해 실제 서비스 가능한 내비게이션 시스템 구축을 진행할 계획이다.

REFERENCE

- [1] Advanced Traffic Prediction of T-map, <http://readme.skplanet.com/?p=8870> (accessed Jan., 17, 2016).
- [2] K.M. Koo, T. Lee, and H. Park, "Navigation System Using Big Trajectory Data Based on Empirical Knowledge," *Proceeding of the Summer Conference of Korea Information Science Society*, pp. 2150-2152, 2015.
- [3] The Scala Programming Language, <http://www.scala-lang.org/> (accessed Jan., 17, 2016).
- [4] Apache Spark-Lightning-Fast Cluster Computing, <http://spark.apache.org/> (accessed Jan., 17 2016).
- [5] J. Lim, K. Kim, J. Kim, H. Oh, S. Yoon, J. Yoo, et al., "Design and Implementation of a Realtime Optimal Traffic Route Guidance System Through Big Data Analysis," *Proceeding of 2014 Fall Conference of Korea Contents Association*, pp. 297-298, 2014.
- [6] S. Yang, C. Choi, and H. Choi, "Design and Implementation of Vehicle Route Tracking System using Hadoop-Based Bigdata Image Processing," *Journal of Digital Contents Society*, Vol. 14, No. 4, pp. 447-454, 2013.
- [7] K. H. Lee, Y. H. Jo, T. Lee, and H. Park, "Recommendation of Best Empirical Route Based on Classification of Large Trajectory Data," *KIISE Transactions on Computing Practices*, Vol. 21, No. 2, pp. 101-108, 2015.
- [8] W. Sagong, "Design and Implementation of Optimized Route Search Technique based on User Experience Using Open APIs," *Journal of Korea Multimedia Society*, Vol. 18, No. 5, pp. 682-690, 2015.
- [9] GPX: The GPX Exchange Format, <http://www.topografix.com/gpx.asp> (accessed Jan., 17, 2016).
- [10] OpenStreetMap, <https://www.openstreetmap.org> (accessed Jan., 17, 2016).
- [11] OsmAnd: Offline Mobile Maps & Navigation, <http://osmand.net/> (accessed Jan., 17, 2016).
- [12] uMap, <http://umap.openstreetmap.fr/en/> (accessed Jan., 17, 2016).
- [13] H. Karau, A. Konwinski, P. Wendell, and M. Zaharia, *Learning Spark*, O'Reilly Media, 2015.
- [14] M. Odersky, L. Spoon, and B. Venners, *Programming in Scala*, A Comprehensive Step-by-Step Guide, 2nd Edition, Artima Inc, 2011.
- [15] JOSM: Extensible Editor for OpenStreetMap,

<https://josm.openstreetmap.de/> (accessed Jan., 17, 2016).



구 광 민

2010년~현재 상명대학교 컴퓨터 소프트웨어공학과 학사 과정
관심분야: 빅데이터 처리, 인공지능, 기계학습



박 희 민

1993년 서강대학교 전자계산학과 (학사)
1995년 서강대학교 컴퓨터공학과 (석사)
2006년 UCLA 전자공학과 (박사)
2013년~현재 상명대학교 컴퓨터 소프트웨어공학과 조교수
관심분야: 웹기반정보시스템, 빅데이터처리, 사물인터넷



이 태 호

2010년~현재 상명대학교 컴퓨터 소프트웨어공학과 학사 과정
관심분야: 빅데이터 처리, 웹 기술, 응용소프트웨어 개발