

# 주변 차량 위치 좌표의 고속 클러스터링을 위한 휴리스틱 알고리즘

최 윤 호\*, 유 승 호\*, 서 승 우°

## Heuristic Algorithm for High-Speed Clustering of Neighbor Vehicular Position Coordinate

Yoon-Ho Choi\*, Seung-Ho Yoo\*, Seung-Woo Seo°

### 요 약

분할 계층적 클러스터링(Divisive Hierarchical Clustering)은 하나의 클러스터에서 시작하여 각각의 데이터가 독립된 클러스터에 속할 때까지 각 클러스터를 분할하고 분할된 클러스터 간에 데이터를 이동하는 과정을 반복 수행한다. 하지만, 이러한 일련의 재귀적 호출 과정에서 입력 데이터가 임의적으로 선택되는 경우, 클러스터 내 데이터의 많은 이동을 야기할 수 있다. 이로 인해 주변 차량의 위치를 추정하여 수집된 위치 좌표 정보를 고속으로 클러스터링 할 필요가 있는 로컬 맵 생성 과정에서 사용하기 어렵다는 단점이 있다. 본 논문에서는 주변 차량 위치 추정 과정에서 차량의 주행 방향 정보를 활용하여 분할된 클러스터를 구성하는 데이터의 임의성을 제거함으로써, 클러스터링 연산 속도를 평균 40% 가량 향상시킬 수 있는 새로운 고속의 분할 계층적 클러스터링 방법을 제안한다.

**Key Words** : V2V communication, vehicle localization, heuristic, divisive, clustering

### ABSTRACT

Divisive hierarchical clustering algorithms iterate the process of decomposition and clustering data recursively. In each recursive call, data in each cluster are arbitrarily selected and thus, the total clustering time can be increased, which causes a problem that it is difficult to apply the process of clustering neighbor vehicular position data in vehicular localization. In this paper, we propose a new heuristic algorithm for speeding up the clustering time by eliminating randomness of the selected data in the process of generating the initial divisive clusters.

### I. 서 론

차량 간 통신 기술은 신호 또는 데이터를 송수신하는 무선 통신을 활용하여 주행 혹은 정지 중인 차량들

간에 정보 교환이 가능하도록 하는 기술으로써, 이를 활용하여 지능형 자동차를 구현하기 위한 응용 연구가 활발히 진행되고 있다. 특히 차량의 주행성과 안정성을 향상시키는 것을 목적으로 다양한 연구가 활발하

※ 본 연구는 한국연구재단 논문연구과제(NRF-2013R1A1A1005991) 지원으로 수행되었습니다.

◆ First Author : CS, Kyonggi University, ychoi@kyonggi.ac.kr, 정회원

° Corresponding Author : EECS, Seoul National University, sseo@snu.ac.kr, 종신회원

\* Tiberio, seungho\_yoo@tiberio.com

논문번호: KICS2014-01-015, Received January 27, 2014; Revised March 28, 2014; Accepted April 11, 2014

게 진행되어 왔다<sup>11)</sup>.

미국, 유럽, 일본 그리고 한국의 자동차 기업들은 802.11 표준화 위원회에서 승인된 차량 간 무선 통신을 위한 Wireless Access in Vehicle Environment (WAVE) 표준<sup>12)</sup>을 적용한 무인 주차 및 주행 시스템, 주행 중 차량 충돌 방지 시스템 등의 다양한 응용 분야에서 주목할 만한 연구 성과를 보이고 있다. 이처럼 차량의 주행성과 안전성을 향상시킬 수 있는 시스템을 개발하기 위해서는 주변 차량들에 대한 위치 좌표 추정 기술인 로컬 맵 생성에 대한 연구가 반드시 필요하다<sup>13)</sup>.

로컬 맵 생성 과정에서 차량 간 통신을 활용하는 경우는, 단일 차량에서 획득 가능한 정보의 한계를 벗어나 자차 및 주변 차량에서 수집된 많은 정보를 활용함으로써 주변 차량 위치 좌표를 보다 정확하게 추론하는 것이 가능하다<sup>14,15)</sup>. 즉, 하나의 차량에 장착된 센서가 측정하기 어려운 음영 지역에 있는 차량의 위치 정보를 추정하거나, 특정 차량에 대해 주변 차량이 측정한 위치 정보를 활용하여 해당 차량의 위치 추정 시 정확도를 향상시킬 수 있다.

로컬 맵 생성 과정은 그림 1에 기술된 바와 같이 자료 수집 및 가공 단계와 차량 위치 추정 단계로 나뉜다. 자료 수집 및 가공 단계는 다시 자차에서 주변 차량에 대해 수집한 위치 좌표의 공간 동기화 과정과 주변 차량으로부터 수신한 위치 정보와 자차에서 수집되어 가공된 위치 정보의 시간 동기화 과정으로 구성된다. 또한, 차량 위치 추정 과정은 수집된 위치 정보의 클러스터링 과정과 클러스터로부터 차량의 정확한 위치 좌표를 추정하는 차량 위치 근사화 과정으로 구성된다.

그림 1의 자료 수집 단계에서 자신의 주변 차량에 대해 수신한 여러 위치 정보 값은 수집 시 사용되는 차량의 GPS와 센서의 성능 및 도로 지형 그리고 기후 등에 따라 동일한 차량에 대해서도 차량 별로 상이한 값을 가지게 된다. 따라서 자차에서 수집된 데이터와 주변 차량으로부터 수신된 분류되지 않은 데이터

값을 주변 차량별로 분류하는 작업이 필요하다. 이와 같이, 자차 및 주변 차량들로부터 수집된 시간적, 공간적으로 동기화된 데이터를 주변 차량 별로 분류하기 위해 데이터 클러스터링이 사용된다.

데이터 클러스터링은 크게 평면적 클러스터링과 계층적 클러스터링의 두 가지로 나뉜다. 평면적 클러스터링<sup>19,12)</sup>은 일반적으로 클러스터의 수를 알아야 클러스터링이 가능하며, 개선된 알고리즘의 경우도 현실적으로 적정 값을 찾기 어려운 인자들에 민감하다는 단점으로 인해 차량의 위치 좌표를 클러스터링 하는 경우에 사용하기 어려운 문제점을 안고 있다. 계층적 데이터 클러스터링<sup>13,14)</sup>은 각 요소들로부터 시작한 클러스터들이 나무 구조를 생성하는 병합 계층적 클러스터링과 분할 계층적 클러스터링이 주로 사용된다. 하지만, 병합 계층적 클러스터링 역시, 초반의 잘못된 클러스터링이 결과에 큰 영향을 미치는 단점이 있어 애드-혹(ad-hoc) 환경에서 동작하는 차량 간 통신으로부터 수집되어 GPS 및 센서 탐지 오류 등의 다중 오류를 포함하는 데이터의 클러스터링에 적용하기는 적절하지 않다.

분할 계층적 클러스터링<sup>13)</sup>은 하나의 클러스터에서 시작하여 각각의 데이터가 독립된 클러스터에 속할 때까지 분할과 분할된 클러스터 간 데이터 이동을 재귀적으로 반복 수행하는데 이때, 클러스터 간 거리가 가장 멀어지도록 데이터를 분할하여 클러스터 계층을 형성한다. 하지만 분할된 클러스터를 생성하는 재귀적 호출 과정에서 임의의 데이터를 선택하는 과정은 임의성을 활용한 휴리스틱이기 때문에 분할된 클러스터 간 데이터의 많은 이동을 야기할 수 있다.

본 논문에서는 이러한 분할 계층적 클러스터링의 문제점을 해결하기 위하여 분할 클러스터의 데이터를 선택하는 과정에서 차량의 주행 방향 정보를 활용함으로써, 클러스터 간 이동하는 데이터의 양을 줄이고 이를 통해 재귀적 클러스터링 호출 횟수를 감소시키는 새로운 휴리스틱 알고리즘을 제안한다. 즉, 분할 클러스터를 생성하는 과정에서 클러스터를 구성하는 데이터의 임의성을 제거함으로써 분할 계층적 클러스터링의 속도를 향상시킨다.

본 논문의 의의를 요약하면 다음과 같다. 첫째, 최초로 자동차 주행 방향과 위치 정보를 결합한 분할 계층적 클러스터링 알고리즘을 제안한다. 둘째, 차량 간 통신을 활용한 로컬 맵 생성 과정에서 분할 계층 클러스터링 알고리즘의 클러스터링 연산 속도를 평균 40% 가량 향상시킨다. 이는, 수집된 전체 입력 데이터에 대한 연산을 반복적으로 수행하여 로컬 맵 생성 과정

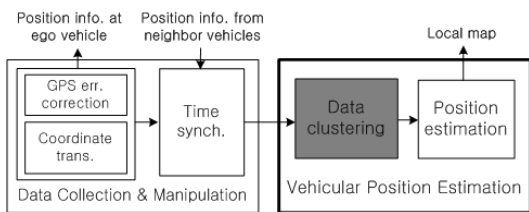


그림 1. 로컬 맵 생성 과정 개략  
Fig. 1. Overview of local map generation

의 연산 속도를 좌우하는 클러스터링을 고속화하고 이를 통해, 차량의 고속 주행 상황에서 필요한 고속의 로컬 맵 생성을 가능하게 한다.

본 논문은 다음과 같이 구성되어 있다. 먼저 단락 II에서는 로컬 맵 생성 과정에 대해서 관련 연구 결과를 중심으로 설명하고, 단락 III에서는 데이터 클러스터링 알고리즘의 특성과 기존의 분할 계층적 알고리즘이 갖고 있는 문제점에 대해 기술한다. 단락 IV에서는 제안된 알고리즘의 상세 동작을 설명하고, 단락 V에서는 그 성능 검증 결과를 기술한다. 마지막으로, 단락 VI에서는 검증 결과를 바탕으로 본 논문을 요약한다.

## II. 로컬 맵 생성 개요

본 단락에서는 그림 1에서 기술된 로컬 맵 생성 과정의 단계 별 동작에 대하여 간략하게 기술한다.

### 2.1 자차에서 수집된 위치 좌표의 공간 동기화

#### 2.1.1 GPS 오류 보정

GPS는 위성 항법 시스템으로 절대적인 위치와 시간을 제공한다. 차량 위치 정보를 추정하기 위한 클러스터링 과정이 높은 정확도를 가지고 클러스터링 되기 위해서는 GPS 정보가 각 차량을 구분할 수 있는 수준의 정확도를 제공해야 한다. 차량과 차선의 너비는 각각 대략 5 미터, 3미터 정도이기 때문에 주행 중인 차량을 구분하기 위해서 요구되는 위치 오차는 3미터 이하로 정의할 수 있다<sup>[6]</sup>. 그러나 차량용으로 사용되는 저가형 GPS의 경우 대략 10 미터 정도의 Circular Error Probability(CEP) 오차를 나타낸다. 따라서 NTRIP(Networked Transport of RTCM via Internet Protocol)<sup>[7]</sup> 등의 알고리즘을 사용하여 오차를 보정한 GPS 정보를 사용한다.

#### 2.1.2 센서 탐지 차량 위치 정보의 공간 동기화

차량의 센서를 이용해 측정한 주변 차량의 위치 좌표 값을 이용하기 위해서 감지한 주변 차량의 위치와 위치 변화량을 측정 차량 기준의 직교 좌표(cartesian coordinate)로 변환한다. 좌표 변환을 위해서는 탐지하고자 하는 차량의 주행 방향과 절대 좌표가 필요하다. 차량의 절대 좌표는 GPS를 통하여 획득 가능하고 차량의 주행 방향은 GPS 데이터의 추적 혹은 Yaw Rate 센서<sup>[8]</sup> 등을 활용하여 획득 가능하다. 탐지한 차량의 방향과 절대 좌표를 알게 되면 차량에 장착된 센

서를 이용해 측정한 주변 차량의 위치 좌표 값을 직교 좌표로 변환한다.

### 2.2 자차와 주변 차량들로부터 수집된 위치 정보의 시간 동기화

각 차량은 차량 간 통신을 이용하여 주변 차량에서 수집 및 가공된 차량 위치 정보를 수신한 후, 수집된 위치 정보의 시간을 동기화 시킨다. 이는 각 차량의 센서가 서로 다른 탐지 주기와 오프셋을 가지고 있고, 차량 간 무선 통신을 통해 주변 차량들로부터 전송된 위치 정보에 전송 지연 시간 등이 추가되기 때문이다. 따라서 각 차량에서 로컬 맵을 작성하는 시간을 기준으로 수신한 위치 정보의 시간 동기화를 수행해야 한다.

### 2.3 데이터 클러스터링

자료 수집 및 가공 단계에서 시간적, 공간적으로 동기화 된 데이터는 몇 개의 클러스터로 나누어지는데, 일반적으로 n개의 데이터를 k( $\geq 2$ )개의 클러스터로 나눈다면  $O(k^n)$ 의 경우의 수가 존재하기 때문에 모든 경우를 비교하여 최적의 클러스터를 찾기는 어렵다. 이에 따라, 각각의 주어진 상황에 맞는 차선의 해를 찾는 다양한 휴리스틱 기법들이 연구 개발되었다.

### 2.4 차량 위치 근사화

이 단계에서는 각 차량에 대하여 클러스터링 된 여러 위치 정보로부터 정확한 차량의 위치를 파악한다<sup>[14]</sup>. 일반적으로 이 과정은 관측 값을 바탕으로 측정값의 신뢰도를 추정하거나 동일 차량에서 측정된 데이터들이 공통적으로 가지고 있는 오차를 상쇄시키는 과정, 좌표를 재 추정하여 정확도를 향상시키는 최적화 과정을 포함한다.

## III. 데이터 클러스터링을 위한 휴리스틱 알고리즘

클러스터링 기법은 크게 평면적 클러스터링과 계층적 클러스터링의 두 가지로 나눌 수 있다. 평면적 클러스터링에 속하는 알고리즘은 일반적으로 미리 몇 개의 클러스터로 나누어 질 것으로 예상하고 클러스터링을 수행한다. 그 중 잘 알려진 것으로는 K-means와 그 응용 알고리즘, EM 알고리즘 등이 있다<sup>[9,10]</sup>. 하지만 이들 알고리즘은 일반적으로 클러스터의 수를 알아야 클러스터링이 가능하다는 단점을 가지고 있다. 클러스터 수가 불필요한 DBSCAN<sup>[11]</sup>과 OPTICS<sup>[12]</sup> 알고리즘의 경우, 임의 형태의 클러스터를 찾는 특징이 있으나 현실적으로 적정 값을 찾기 어려운 인자들

에 민감해서 차량의 위치 좌표를 클러스터링하는 경우에 사용하기 어렵다.

계층적 클러스터링에 속하는 알고리즘은 각 요소들로부터 시작한 클러스터들이 나무 구조를 생성하며 leaf 노드에는 각각의 데이터가 있고 root node에는 모든 데이터를 포함하는 클러스터가 위치한다. 이 기법은 계층 생성 시 병합하고 분할하는 방식에 따라 병합 계층적 클러스터링과 분할 계층적 클러스터링으로 구분된다.

병합 계층적 클러스터링의 동작 방식은 먼저 모든 데이터를 각각 클러스터로 만든 다음, 클러스터 중 거리가 가장 가까운 클러스터를 찾아 하나의 클러스터로 병합하고 그 클러스터의 대표 값을 정한다. 이 과정을 하나의 클러스터만 남을 때까지 반복 수행한다. 이 방법은 클러스터 계층이 생겨 데이터를 분석하는데 용이하며 클러스터의 숫자를 알지 못하는 경우에도 클러스터링이 가능하다는 장점이 있다. 반면, 초기에 데이터가 잘못 클러스터링 되는 경우 최종 클러스터링 결과에 큰 영향을 줄 수 있으며 클러스터 간 거리를 측정하는 방식에 크게 의존한다는 단점이 있다.

분할 계층적 클러스터링은 병합 계층적 클러스터링과 유사한 방식으로 동작하지만 하나의 클러스터에서 시작하여 각각의 데이터가 독립된 클러스터에 속할 때까지 분할과 분할된 클러스터 간 데이터 이동이 반복적으로 발생한다는 점에서 차이가 있다. 이 방법은 주어진 데이터로 구성된 하나의 클러스터를 분할하여 클러스터를 생성하고, 생성된 클러스터 간 데이터를 이동하는 과정을 재귀적으로 실행한다. 구체적으로는, 주어진 클러스터로부터 분할하여 생성할 클러스터의 수를 정한 후 클러스터 간 거리가 가장 멀어지도록 계

층을 형성하며, 하나의 클러스터를 분할하여 생성된 클러스터 간에 데이터를 이동하는 동작을 반복하는 방식으로 동작한다.

예를 들어, 그림 2(a)와 같이 4개의 군집을 형성하는 10개의 데이터가 주어지는 경우, 모든 데이터로 구성된 하나의 클러스터를 형성하고 그림 2(b)와 같이 임의로 데이터를 선택하여 두 개의 클러스터로 구성된 나무 모양의 계층 구조를 형성한다. 다음으로, 클러스터간 거리가 가장 멀어지도록 클러스터 간에 데이터를 이동하여 그림 2(c)의 계층적 클러스터를 형성한다. 이러한 일련의 과정을 재귀적으로 실행하여, 그림 2(d)와 같이 4개의 클러스터로 구성된 나무 모양의 계층 구조를 생성한다.

#### IV. 고속의 클러스터링을 위한 휴리스틱 알고리즘

##### 4.1 알고리즘 동작

제안하는 알고리즘은 초기에 동일 차량을 감지한 데이터를 최대한 같은 클러스터에 속하게하여 클러스터 내 데이터 이동을 줄이고 재귀적 클러스터링 호출 횟수를 줄이기 위한 초기 균등 분할 과정(OriDivide()) 과정과 데이터를 분해하고 병합하여 최대 n번의 클러스터링을 수행하여 계층적 나무 형태로 클러스터를 생성하는 과정으로 구성된다. 본 논문에서 사용되어진 용어를 정리하면 다음과 같다.

Algorithm 1. OriDivide(S, De)

```

1: C ← ∅;
2: for all d ∈ S do
3:   if Di==De then
4:     C ← i;
5:   end if
6: end for
7: avg=Ey(C);
8: for all d ∈ C do
9:   if dy < avg then
10:    append d into C1;
11:   else
12:    append d into C2;
13:   end if
14: end for
15: return [C1, C2];
    
```

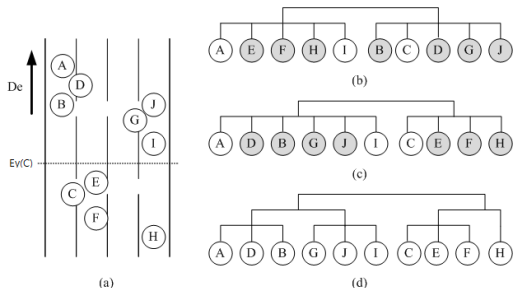


그림 2. 클러스터 분할 과정 비교: (a) 입력 값; (b) 입력 값에 대한 임의의 선택을 통해 분할된 클러스터; (c) 입력 값에 대해 제한된 알고리즘을 적용해 분할된 클러스터 (OriDivide()); (d) 최종적으로 분할된 클러스터  
 Fig. 2. Comparison of cluster division: (a) Input data; (b) Divisive clusters via random selection of input data; (c) Divisive clusters via the proposed clustering algorithm(OriDivide()); (d) Final divisive clusters.

- S: 자차 및 주변 차량으로부터 수집된 차량 위치 정보의 집합
- C: 자차와 차량 주행 방향이 같은 주변 차량으로부터 수집된 좌표 값 클러스터

- $C_i$ :  $i$ 번째 클러스터
- $d$ : 차량 위치의 직교 좌표,  $(dx, dy)$
- $|C|$ : 클러스터  $C$  원소의 개수,  $|C|=n$
- $E(C)$ : 클러스터  $C$ 에 있는 데이터의 평균 좌표 값  $(E(C)=(E(x\text{-좌표}), E(y\text{-좌표})), (x,y) \in C)$
- $E_y(C)$ : 클러스터  $C$ 에 있는 데이터의  $y$ -좌표 평균 값
- $Dist(a, b)$ : 좌표  $a$ 와  $b$ 의 Euclidean distance
- $De$ : 자차의 주행 방향,  $De \in \{0, 1\}$
- $Di$ : 차량  $i$ 의 주행 방향,  $Di \in \{0, 1\}$
- $Th$ : 클러스터링 비용 감소의 경계 값

#### 4.1.1 초기 균등 분할 과정 (*OriDivide()*)

알고리즘 1에서 기술된 바와 같이, *OriDivide()* 함수는 자차에서 수집된 위치 좌표 데이터를 두 개의 클러스터로 분할한다. 분할은 수집된 차량 위치 정보의  $y$ -좌표 평균 값( $E_y(C)$ )을 계산(라인 2에서 7)하여 이를 각 좌표 값과 비교함으로써 자차의 진행 방향의 수직으로 위치 좌표 데이터를 양분(라인 8에서 14)함으로써 수행된다. 자차의 진행 방향에 수직으로 초기 클러스터를 분할하는 이유는 도로 주행 시 측면 차량과의 거리는 전·후방 차량과의 거리에 비해서 매우 가까우므로, 분할 시 동일 차량을 감지한 데이터를 최대한 같은 클러스터에 속하게 하여 클러스터 간 데이터 이동 횟수를 줄이기 위함이다. 또한, 데이터의 평균을 기준으로 하는 이유는 두 클러스터의 크기를 균등하게 분할하기 위함이다.

#### 4.1.2 클러스터링 과정 (*Clustering(C)*)

알고리즘 2에서 기술된 바와 같이, 클러스터링 과정에서는 초기 균등 분할 과정을 통해 생성된 클러스터 각각에 대해 *Clustering()* 함수를 호출하는 과정을 반복적으로 수행한다(라인 18, 19). 알고리즘 2의 첫 번째 *while* 루프(라인 2에서 16)에서 다른 클러스터에 있는 데이터 중에서 클러스터 간 이동 시, 두 클러스터의 대표 값 간 거리를 증가시킬 수 있는 데이터를 이동시킨다. 즉,  $C1$ 에 속한 위치 좌표  $d$ 에 대하여  $Cost(C1, C2) > Cost(C1-d, C2+d)$ 이면, 데이터  $d$ 를  $C2$ 로 이동시킨다. 이때,  $Cost(C1, C2)$ 는 아래 식 (1)과 같이 계산한다.

$$Cost(C1, C2) = 1 / Dist(E(C1), E(C2)) \quad (1)$$

$C2$ 에 속한 데이터에 대하여도 동일한 과정을 수행한다. 이 과정을 비용의 감소가 특정 경계 값  $Th$  이하가 될 때까지 반복(라인 17에서 23)한다.

#### Algorithm 2. Clustering(S, C, De)

```

1: [C1, C2] = OriDivide(S, De);
2: while ΔCost(C1, C2) > ε do
3:   for all d ∈ C1 do
4:     if Cost(C1, C2) > Cost(C1-d, C2+d) then
5:       C1 ← C1 - {d};
6:       C2 ← C2 + {d};
7:     end if
8:   end for
10: for all d ∈ C2 do
11:   if Cost(C1, C2) > Cost(C1+d, C2-d) then
12:     C1 ← C1 + {d};
13:     C2 ← C2 - {d};
14:   end if
15: end for
16: end while

17: if Dist(E(C1), E(C2)) > Th then
18:   Clustering(C1);
19:   Clustering(C2);
20: return [C1, C2];
21: else
22: return [C];
23: end if
    
```

#### 4.2 예제

제안하는 알고리즘의 동작을 설명하기 위하여 그림 3(a)에서와 같이 자차와 같은 방향으로 진행 중인 차량( $De=Di$ )에 대한 9개의 좌표 값이 수집되었다고 가정한다. 9개의 좌표 값은 4대의 주변 차량에 대한 정보로서, 첫 번째 차량에 대해  $\{(1, 13), (1, 12), (2, 13)\}$ , 두 번째 차량에 대해  $\{(5, 10), (7, 10), (6, 8)\}$ , 세 번째 차량에 대해  $\{(2, 6), (1, 5)\}$  그리고, 네 번째 차량에 대해  $\{(7, 3)\}$ 의 정보가 수집되었다고 가정한다. 이렇게 수집된 데이터는 그림 1의 위치 추정 과정을 통해 주변 4대의 차량에 대한 위치 좌표를 정확하게 추정하기 위한 데이터로 사용된다.

먼저, 알고리즘 1을 적용하여 구성된  $C = \{(1, 13), (1, 12), (2, 13), (5, 10), (7, 10), (6, 8), (7, 3), (2, 6), (1, 5)\}$ 를 분할하여 새로운 2개의 클러스터를 생성하기 위한 경계 값인  $avg = (3.556, 8.889)$ 를 얻는다. 즉, 주행 방향  $De$ 에 수직인 방향으로  $C$ 를 2등분하여 그림 3(b)와 같이 클러스터  $C1$ 과  $C2$ 를 생성한다.

그림 3(b)에서  $Cost(C1, C2)$ 를 계산하면 0.163이고  $C2$ 에 속한 임의의 데이터 (6,9)에 대하여  $Cost(C1+(6,8), C2-(6,8))$ 를 계산하면 0.158이므로,  $Cost(C1, C2) > Cost(C1+(6,8), C2-(6,8))$ 이 성립한다. 따라서  $C2$ 에 속한 임의의 좌표 (6,9)를  $C1$ 으로 옮겨 그림 3(c)와

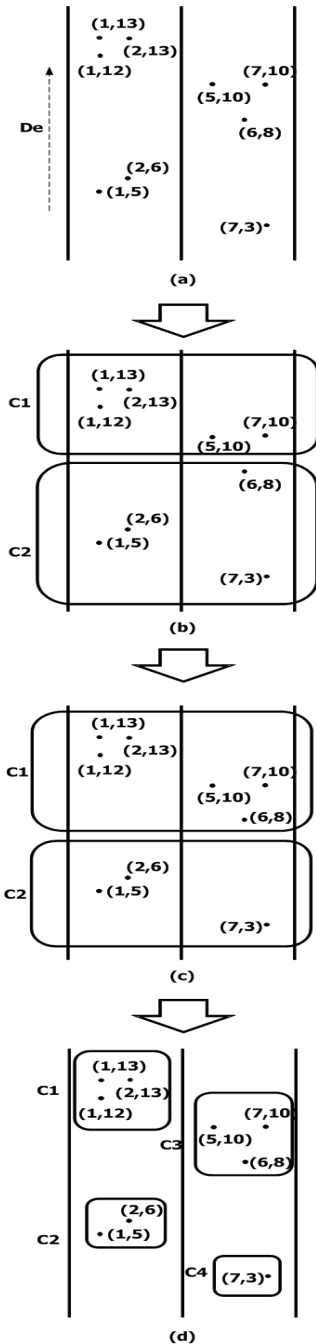


그림 3. 제안한 알고리즘의 동작 예제  
Fig. 3. Operational Example of the Proposed Algorithm

같은 클러스터  $C1$ 과  $C2$ 를 생성한다. 이와 같은 과정을 데이터 이동전의  $Cost(C1, C2)$ 과 데이터 이동 후의  $Cost(C1, C2)$  차이가 즉,  $\Delta Cost(C1, C2)$ 가 충분히 작아져서 데이터 이동이 필요하지 않을 때까지 즉,  $\epsilon$ 에 근접할 때까지 반복 수행한다.

알고리즘 2의 라인 2~16 수행 과정에서 생성된  $C1$ 과  $C2$ 의  $Dist(E(C1), E(C2))$ 가 비용 감소의 경계 값보다 큰 경우,  $C1$ 과  $C2$  각각에 대해서  $Clustering()$  함수를 재귀적으로 호출하여 실행함으로써 그림 3(d)와 같이 4개의 클러스터를 생성한다.

## V. 성능 분석

### 5.1 시뮬레이션 환경

제안된 휴리스틱 알고리즘의 성능을 검증하기 위해서 도로 상에서 차량 간 통신이 가능한 차량과 불가능한 차량이 함께 존재하는 상황을 가정하고, 기존의 분할 계층적 클러스터링 알고리즘과 제안하는 분할 계층적 클러스터링 알고리즘의  $Clustering()$  함수 호출 횟수와 클러스터링 정확도를 측정하여 비교하였다. 입력 데이터 오차 특성의 임의성을 고려하기 위하여 수집 및 분석되는 데이터 좌표 값의 오차 특성은 uniform distribution을 따르는 것으로 가정하였다. 정확도 측정 척도로는 널리 사용되는 F-measure를 이용하였다<sup>[3]</sup>.

### 5.2 시뮬레이션 결과 분석

시뮬레이션 결과는 차량의 통신 장비 장착 비율 별로 5,000번 반복하여 얻은 결과 값의 평균을 계산하여 도출하였다. 그림 4와 5에서 실선은 제안한 알고리즘을 적용한 결과이며, 점선은 기존의 대표적인 분할 계층적 클러스터링<sup>[13]</sup>과 같이 임의 분할 방식을 적용한 결과이다.

그림 4로부터 제안한 알고리즘과 초기 균등 분할 과정을 적용하지 않은 기존의 분할 계층적 클러스터링 알고리즘의 정확도는 유사함을 알 수 있다. 이는

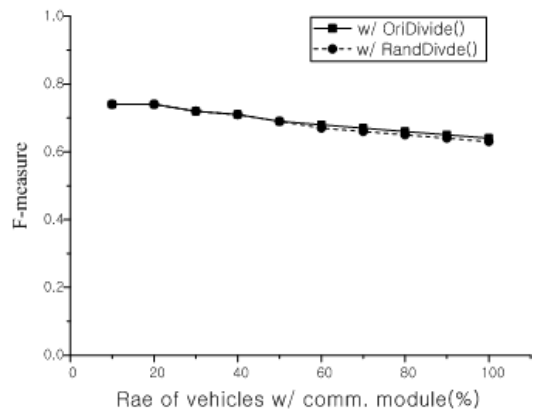


그림 4. 클러스터링 정확성 (F-measure)  
Fig. 4. Clustering accuracy (F-measure)

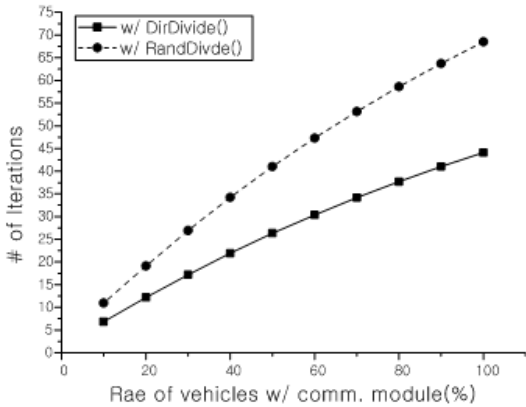


그림 5. 클러스터링 시간 (반복 횟수)  
Fig. 5. Clustering time (# of iterations)

초기 균등 분할 과정을 적용하여 클러스터를 선택하더라도 재귀적으로 *Clustering()* 함수를 호출하고 클러스터 간 데이터를 이동하는 과정을 통해 두 알고리즘이 동일한 클러스터를 형성하기 때문이다.

하지만 제안한 알고리즘은 그림 5에서 나타난 바와 같이, *Clustering()* 함수를 호출하여 하나의 클러스터를 분할하고 분할된 클러스터 간에 데이터를 이동하는 횟수가 기존 알고리즘에 비해 평균 40% 가량 감소한 것을 확인할 수 있다. 이는 분할 계층적 클러스터링의 특성상 클러스터 분할 방식에 따라 클러스터 간에 이동하는 데이터의 수가 크게 변화하기 때문이다. 또한, 이 결과를 통해 제안한 초기 균등 분할 과정을 적용할 경우 차량 주행 환경에서 기존의 많은 분할 계층적 클러스터링 알고리즘들의 클러스터링 속도를 향상시킬 수 있음을 확인할 수 있다. 전체 수집된 입력 데이터에 대한 반복적인 연산을 수행하여 로컬 맵 생성 과정의 연산 속도를 좌우하는 데이터 클러스터링 속도의 향상은 로컬 맵 생성 속도를 향상시킬 수 있음을 나타낸다.

## VI. 결 론

본 논문에서는 차량 간 통신을 활용한 로컬 맵 생성 과정에서 자차 및 주변 차량으로부터 수집된 차량의 위치 정보를 클러스터링하는 과정을 고속화하는 새로운 분할 계층적 클러스터링 방법을 제안하였다. 제안된 알고리즘은 하나의 클러스터를 분할하는 과정에서 차량의 주행 방향 정보를 활용함으로써, 기존의 분할 계층적 클러스터링 알고리즘이 가지고 있는 재귀적 호출 과정에 사용되는 클러스터에 속한 데이터의 임의성을 제거하였다. 그 결과 클러스터 간 이동하

는 데이터의 수와 재귀적 클러스터링 연산 횟수를 감소시켜 클러스터링 속도를 평균 40% 가량 향상시킬 수 있음을 시뮬레이션을 통해 검증하였다. 추후 제안한 알고리즘을 실차에서 구현하여 실제 도로 환경에서 데이터를 측정 및 수집하고, 이를 분석하는 과정을 통해 더 발전된 연구가 가능할 것으로 기대된다.

## References

- [1] R. Parker and S. Valae, "Cooperative vehicle position estimation," in *Proc. IEEE ICC*, pp. 5837-5842, Jun. 2007.
- [2] "IEEE standard for information technology-telecommunications and information exchange between systems-local and metropolitan area network-specific requirements, part11: wireless lan medium access control (MAC) and physical layer (PHY) specification, amendment 6: wireless access in vehicular environment," IEEE Std. 802.11p, 2010.
- [3] A. Boukerche, H. A.B.F. Oliveira, E. F. Nakamura, and A. A.F. Loureiro, "Vehicular ad hoc networks: a new challenge for localization-based systems," *Elsevier ComCom*, vol. 31, no. 12, pp. 2838 - 849, Jul. 2008.
- [4] H. Li and F. Nashashibi, "Cooperative multi-vehicle localization using split covariance intersection filter," *IEEE Intelligent Vehicles Symposium*, pp. 211-216, Jun. 2012.
- [5] S. Fujii, A. Fujita, and T. Umedu, "Cooperative vehicle positioning via V2V communications and onboard sensors," *IEEE VTC*, pp. 1-5, Sept. 2011.
- [6] J. Du and M. J. Barth, "Next-generation automated vehicle location systems: positioning at the lane level," *IEEE T-ITS*, vol. 9, no. 1, pp. 48-57, Mar. 2008.
- [7] G. Weber, D. Dettmering, H. Gebhard, "Networked transport of RTCM via internet protocol (Ntrip) - IP-streaming for real-time GNSS applications," *ION GNSS 18th Int. Tech. Meeting Satellite Division*, pp. 2243-2247, Sept. 2005.
- [8] R. Rajamani, *Vehicle Dynamics and Control*, 2<sup>nd</sup> Ed., Springer Press, 2006.

[9] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," *University of California Press Proc. Mathematical Statistics and Probability*, pp. 281-297, 1967.

[10] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay, "Clustering large graphs via the singular value decomposition," *Machine Learning*, vol. 56, no. 1-3, pp. 9-33, 2004.

[11] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A Density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. KDD*, pp. 226-231, 1996.

[12] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "OPTICS: Ordering points to identify the clustering structure," in *Proc. ACM SIGMOD*, vol. 28, no. 2, pp. 49-60, 1999.

[13] D. Boley, "Principal direction divisive partitioning," *Springer J. Data Mining and Knowledge Discovery*, vol. 2, no. 4, Dec. 1998.

[14] A. Bhawiyuga, H.-H. Nguyen, H.-Y. Jeong, "A fusion of vehicle sensors and inter-vehicle communications for vehicular localizations," *J. KICS*, vol. 37, no. 7, pp. 544-554, Jul. 2012.

**최 윤 호 (Yoon-Ho Choi)**



2002년 8월: 경북대학교 전자  
전기공학부 졸업  
2004년 8월: 서울대학교 전기  
컴퓨터공학부 석사  
2008년 8월: 서울대학교 전기  
컴퓨터공학부 박사  
2010년 12월: 펜실베니아 주립  
대학교 박사후연구원

2012년 2월: 삼성전자 네트워크사업부 책임연구원  
2012년 3월~현재: 경기대학교 융합보안학과 조교수  
<관심분야> 지능형 자동차 IT 보안, 빅 데이터  
분석, 컴퓨터 및 네트워크 보안 성능 최적화

**유 승 호 (Seung-Ho Yoo)**



2010년 2월: 서울대학교 전기  
컴퓨터공학부 졸업  
2013년 2월: 서울대학교 전기  
컴퓨터공학부 석사  
2013년 3월: (주)티베로 선임  
연구원  
<관심분야> 지능형 자동차 IT,  
내장형 소프트웨어

**서 승 우 (Seung-Woo Seo)**



1987년 2월: 서울대학교 전기  
공학과 졸업  
1989년 4월: 서울대학교 전기  
공학과 석사  
1993년 12월: 펜실베니아 주립  
대학교 박사  
1996년 3월~현재: 서울대학교

공과대학 전기컴퓨터공학부 교수  
2009년 3월~현재: 서울대학교 지능형자동차IT 연구  
센터 센터장  
<관심분야> 지능형 자동차 IT, 유무선 통신망 보안  
및 자원 최적화, 스마트그리드