

차량 정보의 실시간 모니터링을 위한 필터링 기반 자동 추출 시스템의 설계 및 구현

김 범 준*, 한 동 현*, 백 윤 주^o

Design and Implementation of Filtering-Based Automatic Extraction System for Real-Time Monitoring of Vehicle Information

Beomjun Kim*, Dongheon Han*, Yunju Baek^o

요 약

차량 기술이 발전함에 따라 차량 내부의 센서들이 늘어나고 있다. 또한 차량 내부 센서 간 통신에 활용되는 CAN 통신에 대한 관심도 함께 높아지고 있다. 하지만, 일부 CAN 메시지들이 표준으로 정의되어 있지 않아 차량 제조사나 차종 별로 달라서 활용이 어려운 문제가 있다. 본 논문에서는 차량의 비표준 정보를 자동으로 추출할 수 있는 시스템을 제안하고 구현하였다. 이 시스템은 스마트폰의 센서 정보를 활용하여 추정된 상황 정보와 CAN 데이터를 분석하여 5종의 CAN 정보를 자동 추출한다. 시스템의 구현을 위하여 CAN 데이터 수집 장치와 스마트폰을 활용한 자동 추출 기법을 개발하였고, 차량 실험을 통해 성능을 검증하였다. 스마트폰 센서를 활용한 상황 추정 기법이 계산한 차량의 이동거리는 오도미터 정보와 94% 유사함을 확인하였고, 가속/감속, 후진, 회전에 대한 정보는 각각 96.8%, 98.3%, 100%, 100%의 높은 추정 정밀도를 나타내었다. 또한 5종의 차량으로 실험한 결과 과반수의 차량에서 자동 추출이 가능함을 확인하였다.

Key Words : Controller area network, OBD device, reverse engineering, Monitoring system

ABSTRACT

As vehicle technology develops, there is a growing interest in CAN communication used for inter-sensor communication. In this paper, we propose and implement a system that can automatically extract vehicle information. Our system analyzes the CAN data and performs automatic extraction of five kinds of vehicle information. We developed automatic extraction method using CAN data collection device, and verified the performance through vehicle experiment. It is confirmed that the calculated moving distance of the vehicle is 94% similar to the odometer information. And, precision of acceleration/deceleration, backward driving, and rotation was 96.8%, 98.3%, 100%, 100%, respectively. In addition, we have experimented with five kinds of vehicles and confirmed that it is possible to automatically extract more than half of the vehicles.

* 이 논문(또는 저서)은 2019년 정부(과학기술정보통신부)의 재원으로 (재)스마트 IT 융합 시스템 연구단(글로벌프런티어사업)의 지원을 받아 수행된 연구임 ((재)스마트 IT 융합시스템 연구단-2011-0031863)

♦ First Author : Pusan National University Dept. of Electrical and Computer Engineering, beomjun.kim@eslab.re.kr, 학생회원

° Corresponding Author : Pusan National University Dept. of Electrical and Computer Engineering, yunju@pusan.ac.kr, 종신회원

* Pusan National University Dept. of Electrical and Computer Engineering, dongheon.han@eslab.re.kr, 학생회원

논문번호 : 201905-060-D-RE, Received April 30, 2019; Revised June 27, 2019; Accepted July 6, 2019

I. 서 론

반도체 기술이 발전함에 따라 차량 내부에 복잡한 네트워크를 형성하는 다양한 센서들이 늘어나고 있다. 차량의 여러 기능들이 디지털화 되면서 네트워크 내부에서 주고받는 데이터의 흐름이 점차 늘어남에 따라 차량용 네트워크에 대한 관심이 높아지고 있다. 그중 차량 내부에 존재하는 CAN에 대한 연구들은 예전부터 지금까지 지속적으로 이루어지고 있다. 이미 많은 제품들이 차량의 고장을 진단하는 목적이나 운전자의 편의를 위한 기능을 제공하기 위하여 CAN 통신을 수행한다. 또한 CAN 통신을 통해 얻을 수 있는 다양한 정보들을 안전이나 운전자의 편의를 위해 활용하는 다른 연구들이 활발하게 진행되고 있다. CAN은 버스 구조를 가지고 있어 여러 구성요소 간의 통신을 위한 비용이 저렴하다는 장점을 가지고 있으며, ID를 기반으로 우선순위를 할당할 수 있는 특징을 가지고 있다.

운전자를 위하여 차량의 정보를 활용하는 다양한 연구들이 존재한다. 사고를 예방하기 위하여 운전자의 피로나 졸음운전을 인식하는 연구¹⁻³⁾가 대표적인 예이다. 또한 운전자의 차선 변경이나 회전을 인식하는 연구⁴⁻⁷⁾도 존재한다. 안전한 운전 습관을 기르기 위하여 운전자의 운전 습관을 분석^{8, 9)}하거나 차량의 도난을 막기 위해 운전자를 구분하는 연구^{10, 11)} 또한 존재한다. 이러한 연구들은 대부분 차량의 정보들을 활용하는데, 표준으로 정의되어 있어 쉽게 구할 수 있는 정보들이 있는가 하면 공개되어 있지 않아 차량 제조사의 협조 없이는 정보를 획득하기 어려운 경우도 많다.

이 문제를 해결하기 위하여 센서를 통해 정보를 얻거나¹²⁾ 차량의 CAN을 분석하는 다양한 기존 연구들¹³⁻¹⁵⁾이 존재한다. 하지만, 기존 연구들이 가지는 한계점으로 인하여 전문가가 아닌 사람들이 차량의 정보를 추출하여 활용하기가 쉽지 않다. 특히 차량 제조사와 차종, 연식에 따라서 달라지는 차량 정보들을 활용하는 방법들이 달라지기 때문에 특정 차종에 대한 정보를 알고 있다고 해도, 다른 차량에 적용하기 어려운 문제가 있다.

본 논문에서는 차량 제조사와 차종마다 달라질 수 있으며 공개되어 있지 않은 문제를 해결하기 위하여 차량용 정보를 자동으로 추출하는 시스템을 제안하였다. 제안하는 시스템에서는 차량의 여러 정보들 중에서 기어 포지션, 브레이크 페달, 스티어링 휠, 오도미터, 방향 지시등 등 5종류에 대한 추출을 시도하

였다. 이를 위해 차량용 CAN 정보를 수집할 수 있는 수집 장치를 설계하고 스마트폰이 무선으로 CAN 데이터를 받을 수 있도록 펌웨어를 구현하였다. 또한 자동 추출 과정을 위하여 스마트폰에 존재하는 센서 정보들을 기반으로 차량의 현재 상황에 대한 추정 기법을 고안하였다. 그리고 추정한 상황 정보와 CAN 데이터를 활용하여 탐승하고 있는 차량에 대한 CAN 정보를 자동으로 추출할 수 있도록 시스템을 구성하였다. 시스템은 추출하고자 하는 차량 정보의 특징을 반영할 수 있도록 구현하였고 성능 평가를 위하여 차량 실험을 수행하였다.

II. 관련 연구

2.1 CAN 통신과 OBD-II

CAN은 Controller Area Network의 약자로 차량 내부에서 각 기능들을 담당하기 위해 존재하는 수많은 전자 제어 장치(Electronic Control Unit, ECU)들이 서로 통신하기 위해 설계된 통신 규격이다. CAN의 대표적인 특징은 별도의 호스트 없이 ECU들이 버스 구조를 공유하여 사용한다는 것이다. 일반적으로 CAN 프레임은 우선순위에 따라서 ID를 11개의 비트 혹은 29비트로 나타낸다. 각 ECU는 CAN 버스가 사용 가능할 때라면 언제든지 메시지 전송이 가능하며, 메시지를 동시에 전송하려고 할 때에는 낮은 ID가 더 높은 우선순위를 가진다. CAN은 차량 진단용 통신 표준인 OBD-II의 대표적인 프로토콜로 활용되고 있다.

온보드 진단기(On-Board Diagnostics, OBD)는 차량의 상태를 진단하여 발생한 고장진단코드(Diagnostic Trouble Codes)의 내역을 확인하거나 차량의 현재 상태를 실시간으로 모니터링하기 위하여 널리 사용된다. 표준에서는 정보를 획득하기 위한 질의응답 과정과 각 정보들의 파라미터 ID 및 표현 형태를 정의하고 있어 정보에 대한 접근이 용이하다. 한국에서도 OBD-II 규격이 의무화됨에 따라 모든 차량이 OBD-II 포트를 가지고 있으며 다양한 용도로 활용되고 있다. OBD-II 포트는 차량용 블랙박스나 네비게이션에서 차량의 정보를 획득하기 위해 활용되기도 하고, 운전자의 편의를 위해 활용되는 전방 시현기(Head-Up Display, HUD)에서도 쓰인다. 그리고 차량의 가속과 감속, 스티어링 휠의 변화와 방향 지시등의 사용 유무는 운전자의 운전 습관이나 안전 운전 점수 등을 추정하는 용도로 사용될 수 있다. 또한 오도미터와 기어포지션, 차량의 현재 위치와 속도 등의 정보들은 차량관제 시스템(Fleet Management System,

FMS)에 활용할 수 있다. 하지만, 차량에서 얻을 수 있는 여러 정보들 중에서 OBD-II에 표준으로 정의되어 있는 정보들은 엔진 RPM, 차량 속도, 스로틀 포지션, 연료 정보 등 일부로 제한되어 있다. 즉, OBD-II 표준에서는 방향 지시등, 오도미터, 스티어링 휠, 기어 포지션, 브레이크 페달 등의 주요 정보들을 정의하지 않고 있다. 각 차량 제조사들은 표준으로 정의되어 있지 않은 정보들을 자유롭게 사용한다. 또한, 차량 제조사가 같더라도 차종 또는 연식에 따라 다른 ID를 사용하기도 한다. 이로 인해, OBD-II 표준으로 정의되지 않은 비표준 정보들을 활용하고 분석하기 위한 다양한 연구들이 이루어지고 있다.

2.2 CAN 분석 및 리버스 엔지니어링 기법

표준으로 정의되지 않은 비표준 CAN 정보들을 분석하고 활용하기 위한 연구들 중 일부는 차량의 ECU를 분해하여 분석하거나 차량의 오동작을 막을 수 있는 환경에서 수행되었다. 이러한 연구들¹⁶⁻¹⁸⁾은 ECU의 동작을 분석하기 위해서 생성 가능한 모든 경우의 CAN 메시지를 전송하는 방식을 사용하였다. 이 방식은 단순히 차량의 동작을 감시하는 것뿐만 아니라 차량의 동작을 제어할 수도 있기 때문에 사고를 예방하기 위한 조치가 필요하다. 또한 CAN 메시지를 만들어 전송한 뒤, 그 반응을 살피기 때문에 분석을 오랜 시간동안 수행해야 하는 문제가 있다.

이런 불편함을 회피하기 위해서, OBD 포트로부터 얻은 데이터를 분석하는 연구들이 수행되었다. 대표적으로는 수집한 CAN 메시지들의 용도와 영역을 구분하는 연구들이 존재한다. CAN 버스에 대한 침입을 감지하기 위한 목적으로 CAN 데이터를 분석한 연구¹³⁾에서는 4개의 클래스를 정의하였다. CAN 데이터들을 고정된 값을 가지는 constant와 정해진 개수의 상태를 오가는 multi-value, 계속해서 증가하는 counter, 차량의 센서 값을 나타내는 sensor로 나누어 분석하였다. 이 연구는 전체 CAN 데이터를 여러 클래스로 구분할 수 있다는 장점을 가지고 있지만, 데이터가 정확히 어떤 기능으로 동작하는지 알 수 없다는 문제점이 있다. 이 연구를 개선하여 리버스 엔지니어링을 용이하도록 만든 연구¹⁴⁾에서는 데이터를 physical value, counter, CRC로 나누어 분석하였다. 하지만, 데이터의 분석은 이 분야의 전문가가 직접 수행하도록 한다는 점에서 한계가 존재한다.

CAN 데이터를 토큰화(Tokenization)하여 분석하기 위한 연구¹⁵⁾는 OBD-II 표준에서 얻을 수 있는 정보와 OBD 포트에서 수집한 CAN 데이터 간의 비교

를 통해 유사한 경향을 보이는 영역을 검출한다. 예를 들어, OBD-II 표준에서 얻은 차량의 속도를 기반으로 데이터를 분석하면, 유사한 패턴을 가지는 바퀴 속도에 대한 정보를 알 수 있다. 하지만, 유사성을 가지는 정보가 무엇인지를 결정하는 방법이 전문가의 경험에 의한 분석에 의존하고 있어 비슷한 한계점이 있다.

특정 기능에 대한 CAN 데이터를 추출하려면 전체 데이터를 분석하여 원하는 기능을 찾는 것이 아니라, 데이터를 필터링하여 원하는 데이터만을 걸러내는 방법이 더 효율적일 수 있다. 필터링 기반의 기존 연구¹⁹⁻²⁰⁾에서는 주어진 시나리오대로 수집한 데이터를 단계별로 필터링하여 목표 CAN ID를 추출하는 방법을 제안하였다. 차량의 기어포지션 정보를 추출하기 위하여 P, R, N, D의 상태에서 데이터를 각각 수집하였고, 방향 지시등의 추출을 위하여 왼쪽만 켜 경우, 오른쪽만 켜 경우, 모두 켜 경우로 나누어 데이터를 수집하였다. 선별 필터 또는 동일 필터에서는 동일한 상태에서는 값이 고정되어야함을 이용하여, 값이 변하는 데이터들을 제외한다. 또한 구별 필터에서는 각 상태별로 데이터가 달라야함을 이용하여, 값이 모두 다른 데이터만을 남긴다. 방향 지시등을 추출하는 과정에서는 비율 필터 또는 교대 필터를 이용하여 깜빡이는 특성을 고려하고 있다. 이 연구는 빠른 시간 내에 효율적으로 특정 기능에 대한 CAN 데이터 분석이 가능하지만, 시나리오를 정확히 따라하는 경우에만 추출이 가능하다는 한계점이 있다. 전문가가 아닌 운전자들은 추출이 정확하게 이루어지지 않는 한계점이 있다.

III. 본 론

3.1 연구의 필요성

차량 내부에 존재하는 여러 정보들을 실시간으로 활용하는 것은 다양한 활용성을 가질 수 있다. 하지만, 이와 같은 정보들은 차량 제조사와 차종, 연식에 따라 달라지기 때문에 일반 사용자들이 접근하기 어려운 문제점이 있다. CAN 프레임의 패턴을 분석하여 토큰 단위로 분리하는 연구들은 데이터를 수집 및 분석하는 과정에서 전문가의 도움 없이는 각 데이터가 어떤 기능을 나타내는지 확인할 수 없다. 또한, 차량을 분해하거나 수많은 CAN 프레임을 활용하여 그 동작을 분석하는 과정은 매우 힘들뿐만 아니라 많은 시간이 소요된다.

CAN 프레임을 실시간으로 분석하기 위해서 제안했던 기존 연구^{19,20)}는 운전자가 시나리오를 숙지한 상태에서 정확하게 정해진 순서대로 차량과 추출 장

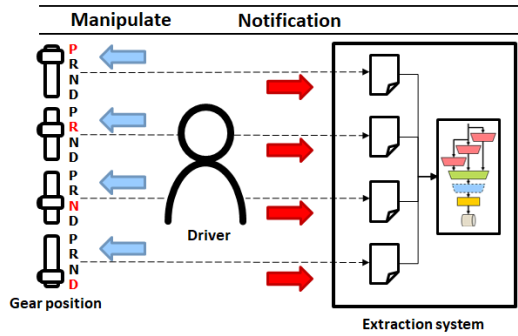


그림 1. 기존의 시나리오 기반 추출 기법
Fig. 1. Existing scenario-based extraction technique

치를 조작해야만 추출이 가능한 문제점이 존재한다. 그림 1에 나타난 기존 기법의 동작 순서를 통해 그 문제를 확인할 수 있다. 기어 포지션에 대한 추출을 수행하려면 기어를 P로 바꾸고, 이를 장치에 알리면 일정 시간동안 데이터가 수집된다. 마찬가지로, 기어를 R, N, D 순서대로 추가적인 데이터를 수집하면 필터링 기반의 추출이 시작된다. 그림에서 나타나는 과정은 하나의 차량에서 하나의 기능에 대한 추출을 수행할 때에 필요한 동작과정이다. 각 기능이 가지는 상태의 수만큼 데이터를 수집해야하며, 사용자는 각 상태의 시작과 끝을 장치에게 알려야 한다. 이 과정에서 다음 상태로 먼저 변경하고 장치의 수집을 종료하거나, 다음 상태로 가기 전에 미리 장치의 수집을 시작하는 조작 실수가 일어날 수 있다. 특히, 데이터를 수집하는 중에 상태가 변화하기 쉬운 브레이크 페달이나 스티어링 휠을 추출하는 것은 익숙하지 않은 운전자가 수집할 경우 한 번에 추출에 성공하기가 어려운 문제점이 있다.

우리는 이 문제를 해결하기 위하여 스마트폰에서 얻을 수 있는 센서 정보를 활용하여 운전자가 자연스럽게 운전하는 과정에서도 자동 추출이 가능한 시스템을 고안하였다. 그리고 각 운전 정보들이 가지는 특성을 활용하여 추출을 진행하는 과정을 추가하였다. 기존 기법에서는 아주 제한적으로 활용하였던 운전 정보들이 가지는 특성을 공통적으로 활용할 수 있도록 하여, 운전자의 개입 없이도 추출이 가능한 기법을 고안하였다. 이 두 가지의 변화로 인하여, 제안하는 시스템은 별도의 시나리오 없이 운전자가 자연스럽게 운전하면서 쌓이는 데이터를 기반으로 주요 CAN 데이터에 대한 자동 추출을 수행한다. 이는 사용자가 차량과 장치를 정확하게 조작하지 못하면 추출이 이루어지지 않았던 기존 기법의 불편함을 해소함으로써

편리성을 증대시킨 사용자 친화적 시스템이다. 즉, 복잡한 과정들을 시스템 내부로 가져와 자동화하고 운전자의 자연스러운 운전 데이터로부터 유용한 정보를 쉽게 추출할 수 있는 장점을 가진다.

3.2 시스템의 설계 및 구성

본 논문에서 제안하는 시스템은 추출을 위한 사용자의 입력을 센서 정보가 대신할 수 있도록 설계하였다. 기존에는 사용자가 추출하려는 비표준 기능의 조작이 변경될 때마다 입력을 통해 알려주던 것을 센서 정보로 대체하도록 하였다.

시스템의 하드웨어는 CAN 수집 장치와 스마트폰으로 나뉜다. CAN 수집 장치는 CAN 트랜시버와 Wi-Fi 또는 BLE 통신이 가능한 시스템 온 모듈 (System on Module, SoM)을 내장하고 있어 OBD 포트에 장착할 경우 CAN 버스에서 주고받는 데이터를 실시간으로 수집하여 스마트폰으로 전송하는 역할을 수행한다. CAN 수집 장치의 구현 결과물은 그림 2의 왼쪽 상단부와 같이 나타난다.

CAN 수집 장치는 수신하는 모든 CAN 프레임의 ID와 payload를 스마트폰에 전송하게 된다. CAN 버스에서 주고받는 프레임의 payload는 최소 1 바이트, 최대 8 바이트로 구성되어 있으며, 스마트폰은 시간 정보를 포함하여 CAN 정보를 저장한다. 저장하는 CAN 관련 정보의 크기는 초당 약 80 KB, 순수한 payload는 초당 약 18 KB 정도로 아주 많은 정보가 수집된다. CAN 데이터 추출 과정에서, CAN 버스에서 프레임을 수집하고 전송하는 역할만을 담당하는 CAN 수집 장치와는 달리 스마트폰이 많은 역할을 담당하게 된다. 이 때, 두 구성요소 간의 대략적인 동작을 그림 2에서 살펴볼 수 있다.

스마트폰은 CAN 수집 장치에서 전송되는 정보에

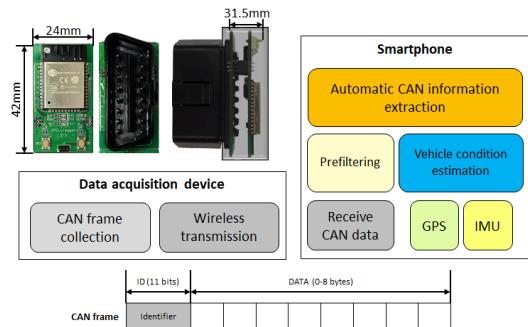


그림 2. 데이터 수집 장치와 스마트폰의 동작과정
Fig. 2. Operation of data acquisition device and smartphone

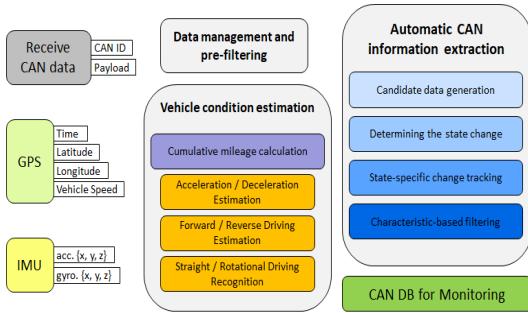


그림 3. 스마트폰을 활용한 자동 추출 과정
Fig. 3. Automatic extraction process using smartphone

시간 정보를 추가하여 저장 및 관리할 뿐만 아니라 센서로부터 수집되는 정보 역시 저장하게 된다. GPS에서는 위도, 경도, 속도, 방향 등을 수집하며, IMU (Inertial Measurement Unit)로부터는 가속도와 자이로 정보를 수집한다. 또한 스마트폰은 수집한 IMU와 GPS 정보를 가공하여 현재 차량의 동작 상황을 추정하며, 이를 활용하여 차량 정보의 자동 추출을 수행한다. 스마트폰에서 수행되는 동작을 자세하게 나타내면 그림 3과 같고, 최종적으로는 추출한 CAN 정보를 스마트폰에 저장하게 된다.

3.3 센서 분석을 통한 상황 추정 기법

제안하는 상황 추정 기법은 실시간으로 수집한 센서 정보를 분석하여 현재 운전 중인 차량이 어떤 상황에 있는지 판단한다. 스마트폰에 내장되어 있는 GPS와 IMU로부터 수집한 데이터를 활용하여 그림 3과 같이 4가지 상황을 추정한다. 먼저, GPS의 위도와 경도, 속도, 시간 정보를 활용하여 차량의 누적주행거리를 계산한다. 누적주행거리는 오도미터의 자동 추출에 활용될 수 있는 상황 정보로 데이터를 수집하기 시작한 위치로부터의 총 주행거리를 계산한 값이다. 그리고 IMU의 가속도 정보와 GPS의 속도 정보를 기반으로 차량의 가속 및 감속 운전을 구분할 수 있다. 정해진 타임 윈도우 크기 TW 동안 센서 정보의 값에 따라 상황을 결정한다. 차량이 가속하고 있을 때는 속도가 선형적으로 증가하며, 차량이 이동하는 방향의 가속도가 평소보다 높은 값을 나타낸다. 반대로 차량이 감속하고 있을 때에는 속도가 선형적으로 감소하며, 차량이 이동하는 방향의 가속도가 평소보다 낮은 값을 유지한다. 유사한 방법으로 후진 운전에 대한 인식이 가능하다. 차량이 후진하기 위해서는 차량의 속도가 0이 되었다가 속도가 증가하는 구간에서 차량이 후진하는 방향의 가속도가 나타나기 때문에 앞으로 주행하는

차량과 다른 패턴이 나타나게 된다. 또한 차량이 좌회전을 하거나 우회전을 하는 경우도 검출할 수 있다. 차량이 속도를 가지고 이동하고 있을 때, 왼쪽이나 오른쪽으로 일정 이상의 가속도가 나타남을 알 수 있다. 센서 정보를 활용한 상황 추정 기법의 전체 동작은 그림 4의 알고리즘과 같이 동작한다.

주행거리는 GPS를 통해 획득한 현재 위치와 이전 위치 간의 거리를 계산해서 얻을 수 있다. 본 논문에서는 위도와 경도 값을 활용하여 직선거리를 계산하는 방법으로 널리 알려진 Haversine 공식^[21]을 사용한다. 이렇게 계산되는 주행거리는 수신되는 GPS 위치의 자체적인 오차, 위치 수집 주기에 따른 오차 등 여러 요인으로 인하여 정확하지 않다. 계산된 누적주행거리와 오도미터를 통해 수집된 누적주행거리의 오차는 그림 5와 같이 나타나며, 약간의 오차가 존재하지만 거의 유사한 것을 알 수 있다. 상황 추정 과정에서 이동 거리를 계산하는 이유는 일정 이상의 거리를 주행함으로써 누적주행거리를 나타내는 오도미터의 값이 변하는 타이밍을 확인하기 위함이다. 그렇기 때문에, 시간이 흐름에 따라 커지는 주행거리의 오차는 중요하지 않다.

GPS와 IMU의 가속도 정보를 활용하는 운전 상황

Algorithm 1 Vehicle Condition Estimation

```

1 : precondition TW, TH_X, TH_Y, TH_R, Speed s[TW],
2 : Acc. x[TW], Acc. y[TW], LatLng preLoc
3 : procedure ConditionEstimate(GPS g, Acc. acc[3])
4 : Speed currentSpeed, LatLng currentLoc, Distance diff
5 : currentSpeed = getSpeed(g)
6 : currentLoc = getLocation(g)
7 : diff = haversine(preLoc, currentLoc)
8 : preLoc = currentLoc
9 : insertSpeed(s, currentSpeed)
10 : insertAcc(x, acc[0], y, acc[1])
11 : if ( isNotZero(s) && ( getAverage(x) < -TH_X
12 : || getAverage(x) > TH_X ) )
13 : vehicleState = TURNING
14 : else {
15 : if (isIncreasing(s) && getAverage(y) > TH_Y)
16 : vehicleState = ACCELERATION
17 : else if (isDecreasing(s) && getAverage(y) < -TH_Y)
18 : vehicleState = DECELERATION
19 : else if ( isIncreasing(s) && getAverage(y) < TH_R )
20 : vehicleState = REVERSE
21 : }
22 : return diff, vehicleState
23 : end procedure
    
```

그림 4. 센서 정보를 활용한 상황 추정 기법의 의사 코드
Fig. 4. Pseudo code of vehicle condition estimation technique using sensor information

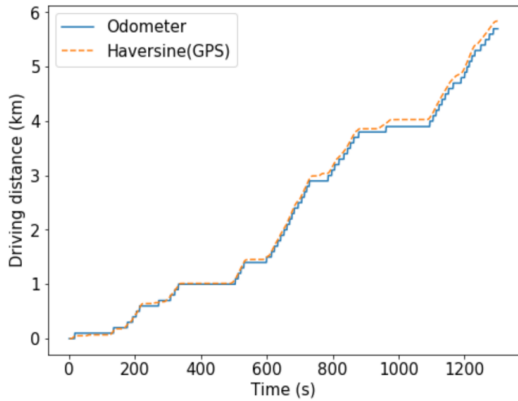


그림 5. GPS와 오도미터를 활용한 주행거리의 비교
Fig. 5. Comparison of mileage using GPS and odometer

에 대한 추정은 경험적으로 얻은 임계값을 기반으로 결정한다. 그림 6과 같이 속도와 가속도 정보를 활용하면 가속, 감속, 후진, 회전하는 상황의 구간을 대략적으로 특정할 수 있다. 우리는 센서 정보를 통해 해당 구간을 결정하는 임계값을 각 상태의 평균보다 큰 값으로 설정했다. 최종적인 목표는 차량 정보의 추출이며, 상황 추정은 그 수단이기 때문이다. 즉, 구간을 정확하게 나누는 정확도보다는 인식한 구간이 정답과 일치하는 정도인 정밀도가 중요하기 때문이다.

3.4 사전 필터링 및 차량 정보의 자동 추출

제안하는 시스템에서는 차량 정보에 대한 자동 추출을 수행하기에 앞서 CAN 데이터에 대한 수집 과정과 사전 필터링 과정을 수행한다. 이 과정은 CAN 자동 추출에 필요한 사전 정보를 수집하면서 미리 데이

터 영역을 줄여놓는 작업이다. CAN 데이터 수집 작업은 차량의 CAN 버스에서 얻을 수 있는 ID와 바이트 위치를 고려하여 데이터를 수집하면서 동시에 센서 분석을 통한 상황 추정 정보를 CAN 데이터와 결합한다. 또한, 사전 필터링은 추출하고자 하는 관심 CAN 정보들은 모두 운전자의 조작에 따라 변하는 값이며 고유한 특성을 가지고 있음을 활용한다. 이 과정에서 지나치게 적거나 많은 상태를 가지는 데이터들을 제외할 수 있다. 또한, 제안하는 시스템의 방향 지시등을 추출하는 과정은 다른 추출 과정과는 다르게 센서 분석을 통한 상황 정보를 활용하지 않는다. 대신 사전 필터링 과정에서 과반수를 차지하는 데이터가 없는 경우를 제외한다. 이는 방향 지시등의 특성인 깜빡임을 이용한다. 운전자가 방향 지시등을 아주 자주 사용한다 하더라도, 켜짐과 꺼짐이 반복적으로 나타나기 때문에 방향 지시등이 꺼진 상태가 과반수를 차지한다. 이와 같이, 사전 필터링 과정은 차량 정보의 자동 추출 과정을 위한 사전 작업들을 수행한다.

5가지 차량 정보의 자동 추출은 수집한 정보들을 최대한 활용하여 필터링을 수행하여 최소한의 후보를 남긴다. 여러 개의 후보가 남게 될 경우, 스코어링을 통해 최종 후보를 선정하게 된다.

먼저, 기어 포지션의 추출 과정은 2가지 상태를 인식하여 활용하고 1가지 상태에 대한 가정을 가지고 있다. 차량이 가속하고 있는 상태의 기어 포지션은 D 이고, 차량이 후진하고 있는 상태의 기어 포지션은 R 이며, 차량의 시동을 걸 때에는 항상 기어 포지션이 P 라고 가정한다. 본격적인 필터링 과정은 기존 기법과 동일하게 수행된다. 수집한 3가지 상태에 대하여, 구

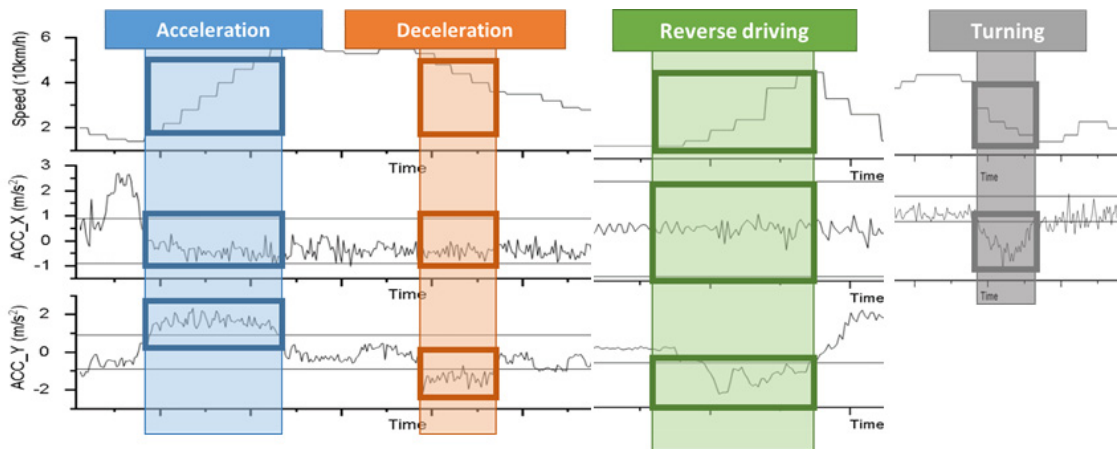


그림 6. 차량의 속도 및 X축, Y축의 가속도 정보를 활용한 차량의 상황 추정
Fig. 6. Vehicle condition estimation using vehicle speed, X-axis acceleration and Y-axis acceleration

간 내에서는 상태가 변하는 데이터는 후보에서 제외한다. 또한, 수집한 3가지 상태의 값이 모두 달라야 한다. 기어포지션의 추출 과정에서 가장 중요한 부분은 후진 데이터를 확보하는 것이다. 자연스러운 운전 데이터에서 후진 데이터가 발생하지 않으면 추출을 수행할 수 없기 때문이다. 이로 인하여, 다른 기능에 대한 추출보다 더 긴 시간이 필요하다는 특징이 있다.

브레이크 페달의 추출은 브레이크 페달을 밟은 상태와 밟지 않은 상태를 각각 인식하여 활용한다. 차량이 감속중일 경우를 브레이크를 밟은 상태로 활용하며, 차량이 가속중일 경우를 브레이크를 밟지 않은 상태로 활용한다. 가속과 감속 상황에 대한 정보는 데이터 수집이 여러 번 일어나기 때문에 2가지 상태만으로도 후보를 특정할 수 있다.

스티어링 휠에 대한 추출은 다른 기능에 대한 추출과는 조금 다르게 수행된다. 활용하는 정보는 차량이 직진하는지 회전하는지에 대한 여부와 상태가 가지는 특성을 고려한다. 스티어링 휠의 데이터는 기어 포지션이나 브레이크 페달과는 다르게 선형적으로 표현된다는 특징을 가진다. 또한, 운전자가 조작하는 스티어링 휠의 각도를 표현하는 값의 변화가 크지 않은 특징을 가지고 있다. 데이터가 전혀 변하지 않거나, 선형적이지 않은 데이터들을 제외하고 나면 후보가 크게 줄어들게 된다. 차량이 회전하는 과정에서 스티어링 휠의 데이터는 증가했다가 감소하거나 감소했다가 증가하는 형태의 데이터를 나타낸다. 이는 회전 후 직진하는 위치로 회귀하게 되는 특성을 반영한 것이다.

오도미터의 추출을 위한 데이터들은 모두 정지한 구간에서 수집하게 된다. GPS를 기반으로 특정 거리 이상 주행한 뒤에 신호 대기로 인하여 정지하였을 때의 데이터를 활용한다. 정지한 상태에서 데이터를 수집하면 수집 과정에서 오도미터의 값이 변하는 것을 막을 수 있다. GPS를 통해 데이터 수집 영역을 결정하는 과정에서 데이터 영역의 앞 뒤 1초를 잘라낸다. 이는 GPS의 속도 정보가 시간적인 오차를 반영한 것이다. 또한, 오도미터 데이터 역시 선형적으로 증가한다는 특징을 가지고 있기 때문에, 이를 추출 과정에 활용한다.

방향 지시등의 추출은 사전 필터링 과정에서 정한 과반수 데이터를 사용하여 수행한다. 방향 지시등 데이터의 변화는 항상 짧은 시간 내에 과반수 데이터로 돌아와야 한다. 이는 방향 지시등이 꺼진 상태가 과반수를 차지하기 때문이다. 이를 활용하여, 과반수 데이터가 없거나 과반수 데이터로 1초 안에 돌아오지 않는 데이터를 모두 제외하는 방법으로 자동 추출이 이

루어진다.

IV. 성능 평가

제안하는 자동 추출 시스템의 성능 평가를 위하여 실험을 다음과 같이 구성하였다. 스마트폰과 무선통신이 가능한 CAN 데이터 수집 장치를 차량에 장착하고 정해진 도로를 주행하였다. 그림 7과 같이 2가지 환경에서 실험을 수행하였다. 도시 환경의 도로의 길이는 약 5km이고, 공원 주차장 실험의 도로 길이는 약 2km이다. 도시 환경의 도로 주행은 길고 복잡한 도로 주행 환경을 가지고 있기 때문에, 방향 지시등과 오도미터 등 2가지 정보를 추출하는 실험에 활용하였다. 또한, 공원 주차장 환경에서는 기어 포지션, 브레이크 페달, 스티어링 휠 등 3가지 정보에 대한 추출을 수행하였다. 실험을 위한 차량은 르노 삼성, 현대, 기아, GM 등의 4개 회사의 차종 5개를 선정하여 실험하였다. 실험의 분석을 위하여 주행 과정을 카메라로 촬영한 동영상 파일과 센서 데이터 또한 로그 파일의 형태로 수집하였다. 우리는 이 실험을 통해 센서 분석을 통한 상황 추정 기법과 차량 정보의 자동 추출에 대한 성능 평가를 진행하였다. 상황 추정 기법의 성능 평가를 위하여 운전 구간에 대한 분석을 수동으로 수행하였고, 이를 잘 인식할 수 있는지 확인하였다.

또한 차량 정보의 자동 추출은 추출된 CAN 데이터의 분석을 통하여 원하는 정보가 올바르게 추출되었는지를 확인하였다.

GPS를 기반으로 계산한 이동 거리는 전체 데이터에 대하여 그림 8과 같은 유사도를 나타낸다. 약간의 오차를 고려하여 정지 구간 추출 과정에서 앞뒤로 1초씩을 제외하였을 때, 정지 구간의 정밀도가 100%가 됨을 알 수 있었다. 센서 분석을 통한 나머지 4개의

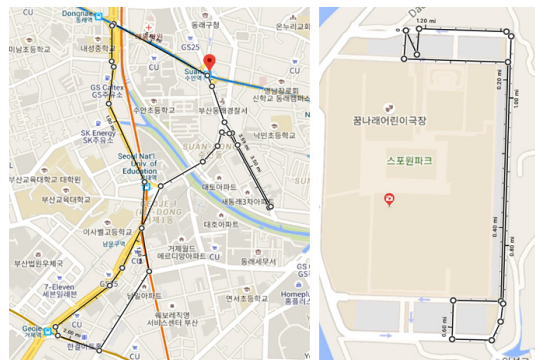


그림 7. 도시 환경과 공원 주차장의 실험 환경
Fig. 7. Experimental environment (city and parking lot)

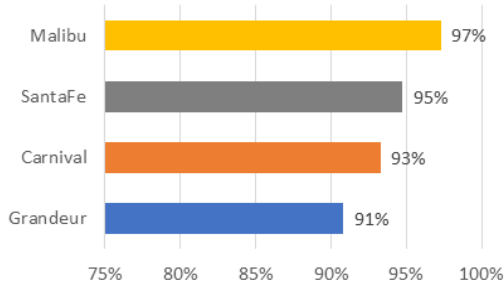


그림 8. 계산한 거리와 오도미터 간 유사도
Fig. 8. Similarity between calculated distance and odometer

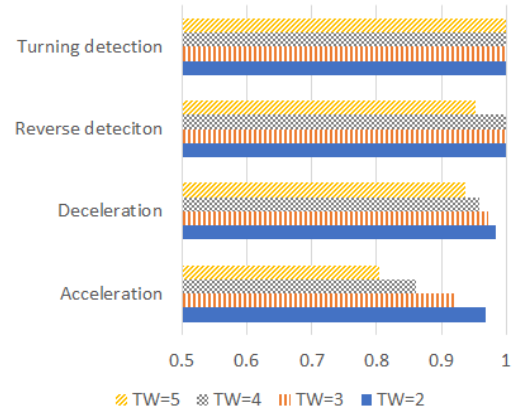


그림 9. 차량 상황 추정 기법의 정밀도 성능
Fig. 9. Precision of vehicle condition estimation

상황 추정의 정밀도 성능은 그림 9와 같이 나타난다. 타임 윈도우의 크기 TW가 2일 때 가장 높은 성능을 나타내며, 가속 인식은 약 96.8%, 감속 인식은 약 98.3%, 후진과 회전 인식은 100%의 성공률을 보였다.

2가지 환경에서 차량 정보의 자동 추출 실험을 수행한 결과는 표 1과 같이 나타난다. 도시 환경에서는 오도미터와 방향 지시등에 대한 자동 추출 실험을 진행하였다. 이 과정에서 오도미터 추출에 필요한 주행 거리를 확인하였다. 또한 시내의 복잡한 도로를 주행 하면서 회전이나 차선을 변경하는 과정에서 필요에 따라 방향 지시등을 사용하였고, 이를 추출할 수 있는지 실험하였다. 5개 차종 중에서 오도미터 정보는 르노 삼성 차량을 제외한 4개가 추출되었고, 방향 지시 등 정보는 기아 차량을 제외한 4개가 추출됨을 확인하였다. 오도미터의 추출은 평균 3.45km의 주행 후 이루어졌으며, 모두 3번 이상의 정지 구간을 인식하여 추출되었다. 이 값은 주행하는 도로의 상황에 따라 달라지는 정지 구간의 패턴에 따라 달라질 수 있다. 방향 지시등의 추출은 평균 3분 3초가 걸렸으며, 주행거리는 약 0.28km를 주행하고 이루어졌음을 확인하였다.

공원 주차장 환경에서는 기어 포지션, 브레이크 페달, 스티어링 휠 등의 3가지 정보에 대한 추출 실험을

수행하였다. 실험 과정에서 공원 주차장 주변을 한 바퀴 주행하고 주차하는 과정에서 자동 추출이 되는지 확인하였다. 우리는 이 실험을 통하여 제안하는 시스템이 3가지 정보를 항상 추출할 수 있음을 확인하였다. 기어 포지션에 대한 평균 추출 시간은 약 3분 9초였으며, 브레이크 페달과 스티어링 휠에 대한 평균 추출 시간은 각각 27.2초와 34.5초였다. 시나리오 기반의 추출이 대부분 30초 이내에 이루어지는 것에 비하면 기어 포지션에 대한 추출 시간은 상대적으로 오래 걸린다. 이는 주차하는 과정에서 후진 데이터가 마지막에 수집되기 때문이다.

V. 결론

본 논문에서는 차량의 CAN을 분석하여 표준으로 정의되지 않은 5종의 차량 정보를 자동으로 추출할 수 있는 시스템을 제안하고 구현하였다. 이 과정에서 정해진 시나리오를 기반으로 필터링을 수행하는 기존 방법과는 다르게 센서 정보를 활용하여 현재 상황을 추정하는 기법을 구현하였다. 정의된 시나리오를 정확히 알고 따라해야 하는 기존 방법과 달리 운전자의 자연스러운 운전 과정에서 차량의 기능을 추출이 가능하다는 점에서 의미가 있다.

제안하는 시스템에서는 스마트폰의 센서 정보를 통해 이동 거리, 가속/감속, 후진, 회전 등 4가지의 상황을 추정하는 방법을 구현하였다. 또한 제안하는 시스템의 성능 평가를 위해 정해진 코스를 주행하여 성능을 확인하였다. 실험을 통하여 차량의 이동거리는 오도미터 정보와 94% 유사함을 확인하였고, 가속, 감속,

표 1. 차량 정보 자동 추출 실험의 결과
Table 1. Results of automatic extraction experiments

Manufacturer	Model	Gear	Brake	Steering	Odometer	Turn signal
Hyundai	Grandeur	O	O	O	O	O
Kia	Carnival	O	O	O	O	-
Renault Samsung	SM6	O	O	O	-	O
Hyundai	SantaFe	O	O	O	O	O
GM	Malibu	O	O	O	O	O

후진, 회전에 대한 정보 추출은 각각 96.8%, 98.3% 100% 100%의 높은 정밀도를 나타냄을 확인하였다.

본 시스템을 활용할 경우, 차량 제조사나 전문가의 도움 없이도 표준으로 지원하지 않는 5종의 차량 운전 정보들을 해석할 수 있다. 특히 기어 포지션과 브레이크 페달, 스티어링 휠, 방향 지시등과 같은 정보들은 운전자의 운전 성향 분석과 운전 상황 인식, 졸음운전 인식 등 여러 분야에서 활용될 수 있다.

References

- [1] K. Vasudevan, A. P. Das, B. Sandhya, and P. Subith, "Driver drowsiness monitoring by learning vehicle telemetry data," *HIS 2017*, pp. 270-276, Ulsan, Korea, July 2017.
- [2] S. Moon, M. Min, J. Nam, J. Park, D. Lee and D. Kim, "Drowsy driving warning system based on gs1 standards with machine learning," *IEEE Int. Congress on Big Data*, pp. 289-296, Honolulu, USA, June 2017.
- [3] Y. Zhang, M. Chen, N. Guizani, D. Wu, and V. C. Leung, "SOVCAN: Safety-Oriented vehicular controller area network," *IEEE Commun. Mag.*, vol. 55, pp. 94-99, Aug. 2017.
- [4] A. Sathyanarayana, S. O. Sadjadi, and J. H. Hansen, "Leveraging sensor information from portable devices towards automatic driving maneuver recognition," *15th Int. IEEE Conf. Intell. Transp. Syst.*, pp. 660-665, Anchorage, USA, Sep. 2012.
- [5] Y. Zheng and J. H. Hansen, "Lane-change detection from steering signal using spectral segmentation and learning-based classification," *IEEE Trans. Intell. Veh.*, vol. 2, pp. 14-24, Mar. 2017.
- [6] V. Leonhardt and G. Wanielik, "Recognition of lane change intentions fusing features of driving situation, driver behavior, and vehicle movement by means of neural networks," *Advanced Microsystems for Automotive Applications*, pp. 59-69, Aug. 2017.
- [7] V. Leonhardt and G. Wanielik, "Feature evaluation for lane change prediction based on driving situation and driver behavior," *20th Int. Conf. Inf. Fusion*, pp. 1-7, Aug. 2017.
- [8] S. H. Chen, J. S. Pan, and K. Lu, "Driving behavior analysis based on vehicle OBD information and AdaBoost algorithms," *IMECS 2015*, pp. 18-20, Hong Kong, Mar. 2015.
- [9] J. E. Meseguer, C. K. Toh, C. T. Calafate, J. C. Cano, and P. Manzoni, "Drivingstyles: A mobile platform for driving styles and fuel consumption characterization," *J. Commun. and Netw.*, vol. 19, pp. 162-168, Apr. 2017.
- [10] M. Enev, A. Takakuwa, K. Koscher, and T. Kohno, "Automobile driver fingerprinting," in *Proc. Privacy Enhancing Technol.*, vol. 2016, pp. 34-50, Jan. 2016.
- [11] B. I. Kwak, J. Woo, and H. K. Kim, "Know your master: Driver profiling-based anti-theft method," *14th Annu. Conf. Privacy, Secur. and Trust*, pp. 211-218, Auckland, New Zealand, Dec. 2016.
- [12] S. Lawoyin, D. Y. Fei, and O. Bai, "Accelerometer-based steering-wheel movement monitoring for drowsy-driving detection," in *Proc. Institution of Mechanical Eng., Part D: J. Automob. Eng.*, vol. 229, no. 2, pp. 163-173, Apr. 2014.
- [13] M. Markovitz and A. Wool, "Field classification, modeling and anomaly detection in unknown CAN bus networks," *Veh. Commun.*, vol. 9, pp. 43-52, Mar. 2017.
- [14] M. Marchetti and D. Stabili, "READ: Reverse engineering of automotive data frames," *IEEE Trans. Inf. Forensics and Secur.*, vol. 14, no. 4, Apr. 2019.
- [15] M. E. Verma, R. A. Bridges, and S. C. Hollifield, "ACTT: Automotive CAN tokenization and translation," arXiv preprint arXiv:1811.07897, pp. 1-6, Nov. 2018.
- [16] T. Hermans, J. Denil, P. De Meulenaere, and J. Anthonis, "Decoding of data on a CAN powertrain network," *16th Annu. Symp. on Commun. and Veh. Technol. in the Benelux*, vol. 6, 2009.
- [17] K. Koscher, A. Czeskis, F. Roesner, S. Patel, and T. Kohno, "Experimental security analysis of a modern automobile," *IEEE Symp. Secur.*

and Privacy, pp. 447-462, Oakland, USA, May 2010.

- [18] S. Freiburger, M. Albrecht, and J. Käufel, "Reverse engineering technologies for remanufacturing of automotive systems communicating via CAN bus," *J. Remanufacturing*, p. 6, Dec. 2011.
- [19] J. Lim, D. Han, Y. Jeon, and Y. Baek, "Design and implementation of in-vehicle information extraction device for advanced driver assistance systems," in *Proc. Symp. KICS*, pp. 351-352, Jun. 2017.
- [20] J. Lim, D. Han, Y. Jeon, and Y. Baek, "ID extraction OBD-II vehicle operation information for driver behavior analysis," in *Proc. Symp. KICS*, pp. 100-101, Jan. 2018.
- [21] M. Zdeb, "Driving Distances and Times Using SAS® and Google Maps," *SAS Global Forum*, vol. 2010, 2010.

김 범 준 (Beomjun Kim)



2012년 2월 : 부산대학교 컴퓨터공학과 공학학사
2014년 2월 : 부산대학교 컴퓨터공학과 공학석사
2014년 3월~현재 : 부산대학교 컴퓨터공학과 박사과정

<관심분야> 지능형 자동차, 저전력 기술, 임베디드 시스템

[ORCID:0000-0002-2945-1054]

한 동 현 (Dongheon Han)



2017년 2월 : 부산대학교 컴퓨터공학과 공학학사
2019년 2월 : 부산대학교 컴퓨터공학과 공학석사

<관심분야> 임베디드 시스템, 하드웨어 설계, CAN

[ORCID:0000-0002-0431-2568]

백 윤 주 (Yunju Baek)



1990년 2월 : 한국과학기술원 전산학과 공학학사
1992년 2월 : 한국과학기술원 전산학과 공학석사
1997년 2월 : 한국과학기술원 전산학과 공학박사
1999년~2002년 : NHN 기술연구소 소장

2003년~현재 : 부산대학교 컴퓨터공학과 정교수

<관심분야> RFID, WSN, RTLS, 임베디드 시스템

[ORCID:0000-0002-3873-2624]