KIPS Trans. Softw. and Data Eng.
Vol.8, No.12 pp.507~516 pISSN: 2287-5905

https://doi.org/10.3745/KTSDE.2019.8.12.507

A Design and Implementation of Floor Detection Application Using RC Car Simulator

Yoona Lee† · Young-Ho Park†† · Sun-Young Ihm†††

ABSTRACT

Costs invested in road maintenance and road development are on the rise. However, due to accidents such as portholes and ground subsidence, the risks to the drivers' safety and the material damage caused by accidents are also increasing. Following this trend, we have developed a system that determines road damage, according to the magnitude of vibration generated without directly intervening the driver when driving. In this paper, we implemented the system using a remote control car (RC car) simulator due to the limitation of the environment in which the actual vehicle is not available in the process of developing the system. In addition, we attached a vibration sensor and GPS sensor to the body of the RC car simulator to measure the vibration value and location information generated by the movement of the vehicle in real-time while driving, and transmitting the corresponding data to the server. In this way, we implemented a system that allows external users to check the damage of roads and the maintenance of the repaired roads based on data more easily than the existing systems. By using this system, we can perform early prediction of road breakage and pattern prediction based on the data. Further, for the RC car simulator, commercialization will be possible by combining it with business in other fields that require flatness.

Keywords : Real-Time Recognition, Floor Detection, Road Breakage, Automatic Driving

RC카 시뮬레이터를 이용한 바닥 탐지 응용 설계 및 구현

이 유 나† · 박 영 호†† · 임 선 영†††

요 약

도로 보수 및 도로 개발에 투자되는 비용이 증가세에 놓여있다. 그러나 포트 홀이나 지반 침하와 같은 사고들로 인하여 운전자들의 안전에 대한 위험성과 사고들로 인해 발생하는 물질적인 피해 역시 증가하고 있다. 이러한 추세에 따라 주행 시 발생하는 진동의 크기에 따라 운전자가 직접적인 개입 없이 도로 파손 여부를 판단하기 위한 시뮬레이션 시스템을 개발했다. 본 논문에서는 시스템을 개발하는 과정에서 실제 차량을 사용할 수 없는 환경의 제한으로 인하여 RC카 (Remote Control Car, 이하 RC카) 시뮬레이터를 사용하여 시스템을 구현하였다. 또한, RC카 시뮬레이터 차체에 진동 센서와 GPS 센서를 부착하여 주행하는 동안 실시간으로 차량의 움직임으로 발생하는 진동 수치와 위치 정보를 측정, 해당 데이터들을 서버로 전달하였다. 이로서 외부 사용자가 데이터를 기반으로 도로 파손 여부와 보수가 진행된 도로의 점검을 기존 방법보다 용이하게 파악할 수 있도록 응용을 구현하였다. 본 논문에서 설계 및 구현한 시스템을 통하여 향후 도로 파손에 대한 조기 대처 및 데이터를 기반으로 패턴 예측을 할 수 있을 것이며, RC카 시뮬레이터의 경우 평평도가 요구되는 다른 분야의 사업과 접목시켜 상용화가 가능할 것으로 예상된다.

키워드 : 실시간 인식, 바닥 탐지, 도로 파손, 자동 주행

1. 서 론

국토교통부의 도로 현황 통계에 따르면 2018년 우리나라의 전체 개통 도로의 93%는 포장도로로 구성되어 있다. 또, 지난 5년간 국토교통부의 도로보수 실적 통계에 의하면 2014년부터 가장 최근에 측정된 2018년도까지 포장 도로 및 위험 도로 개선에 사용되는 비용들은 매년 증가추세를 보이는 현황이다[8].

도로 개선에 지속적으로 투자가 되고 있음에도 불구하고 싱크 홀과 같은 지반 침하로 인하여 일어나는 사고들에 대해서 다양한 매체(중앙일보, 2018.9.12; 한겨레, 2018.7.11)를 통해 꾸준한 보도가 있었다. 더 나아가, 2018년 8월에는 집중호우로 지반이 약해지면서 전국 곡곡에서 '싱크 홀(sink hole)'이 하루에만 평균 2.6개가 발생하여 행인이나 운전자들

※ 이 논문은 2019년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임(No.2018-0-00225, 과학적 정책 수립을 위한 도시행정 디지털트윈 핵심 기술 개발).
※ 이 논문은 2018년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. NRF-2018R1D1A1B07046550).
† 비 회 원 : 고려대학교 컴퓨터학과 석사과정
†† 종신회원 : 숙명여자대학교 IT공학과 교수
††† 준 회 원 : 숙명여자대학교 빅데이터활용 연구센터 책임연구원
Manuscript Received : September 26, 2019
Accepted : October 22, 2019
* Corresponding Author : Sun-Young Ihm(sunnyihm@sm.ac.kr)

이 발견하지 못하여 사고가 일어나는 경우가 빈번하였다[1].

최근, 도로파손으로 발생하는 재난을 줄이기 위해 각 지자체에서 다양한 노력을 기울이고 있다. 서울시에서는 다채널 투과 레이더(3D Ground Penetrating Radar, GPR) 및 도로지반조사차량(Road Survey Vehicle, RSV)등을 도입하여 서울시 내 동공 탐사를 수행함과 동시에 자체적으로 개발한 포장 상태 지수 (Seoul Pavement Index, SPI)를 적용하고 있다. 그 외에는 고속도로의 감시 및 순찰 체계를 강화하여 유지보수를 실시하거나, 포트 홀을 자동 탐지하도록 하는 인공 지능(Artificial Intelligence, AI) 기술 중 딥 러닝(Deep Learning) 기법을 적용하여 신기술을 개발하고 있다[2].

해외 사례의 경우 일본에서는 교각이나 도로, 주변에 '고속도로 스캐너 차량'을 사용하여 도로에 대한 입체적인 이미지를 스캔, 인공 지능을 사용하여 도로의 변형이나 파손을 찾아내고, 교통량이나 과거의 점검 기록을 통해서 보수를 진행하는 서비스를 2017년에 투입하였으며, 미국의 경우 실제 차량에 최첨단 기기를 장착, 실제 주행을 함으로서 실시간으로 도로 파손 정보를 얻어오고 있다.

이러한 노력에도 불구하고 도로 파손은 꾸준히 제기되고 있는 문제라는 점을 통해서 앞서 제시된 방법들은 지속성에 한계가 있다는 것을 유추할 수 있다. 그러나 도로 파손은 당장 직접적인 문제를 야기할 뿐만 아니라, 적절한 보수 조치가 이뤄지지 않을 경우 앞서 언급한 사고들처럼 2차 사고로 이어질 수 있기 때문에 도로 파손과 보수 이후의 점검과 관련된 시스템의 개선이 필요한 상황이다.

이와 관련하여 도로 보수 이후에도 같은 문제가 재발하는지 파악하는 등의 지속 모니터링 및 사전에 포트 홀, 싱크 홀이나 지반 침하와 같은 재난에 대한 피해를 사전 감지할 수 있는 시스템의 필요성을 제기하게 되었다. 따라서 본 논문은 이에 따라 다양한 종류의 센서를 활용한 효과적인 도로 파손 탐지 시뮬레이터를 설계하고 구현하는 방안을 제안한다.

본 논문에서 제안하는 도로 파손 탐지 시뮬레이터는 프로토타입으로 제작된 RC카에 진동 센서와 GPS 센서 같은 다양한 센서를 부착한 상태로 주행을 한다. 실시간으로 측정되는 진동의 수치를 통해 주행하는 해당 지역의 도로의 파손 정도를 탐지한다. 본 논문의 공헌은 다음과 같다.

- 도로 파손을 미리 인지하기 위한 시뮬레이터를 설계함으로써 도로 파손으로 인한 인재 및 사고를 사전에 방지하는데 활용할 수 있도록 한다.
- 데이터베이스에 저장된 데이터들을 통해서 패턴 예측 모델을 설계하여 이를 기반으로 더 체계적인 관리가 가능하도록 한다.
- RC 카 시뮬레이터의 경우 보도블록이나 실내 바닥의 평평도가 요구되는 다양한 분야와 접목시킬 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 기존에 비슷한 시스템 및 연구와 비교 및 분석을 통하여 제안하는 시스템의 특징과 기능을 설명한다. 3장에서는 시스템의 개요 및 주요 기능, 개발 환경을 기술하고, 4장에서는 위에서 설명한 기능들을 구현한 알고리즘 등과 같은 구체적인 방법을 기술한다. 마지막으로 5장에서는 결과와 기대효과를 설명함으로서 본 논문의 결론을 맺는다.

2. 관련 연구 및 기존 서비스 분석

본 장에서는 모바일 애플리케이션, 센서를 사용하여 도로 파손을 탐지 및 관리하는 연구와 서비스를 분석한다. 이러한 분석을 통해 제안하는 시스템과의 비교를 통해서 본 논문에서 제안하고자 하는 시스템의 특징을 소개한다.

2.1 도로 파손 탐지 애플리케이션

이미지 처리 기술을 활용한 도로의 손상 및 파손을 감지하는 연구는 활발하게 진행되고 있다. 객체 기반의 도로 파손 탐지 연구에서는 건물 붕괴의 잔해로 인한 도로 차단을 위하여, 파손을 감지하는 연구를 수행하였다[9]. 또한, 지진 이후의 도로 파손을 감지하기 위해 [10]에서는 영상의 도로 벡터 데이터를 중첩하여 탐지 방법을 적용하였다.

대부분의 도로 파손 탐지 관련 연구들은 도로에 파손 탐지 정확도를 높였지만, 파손 존재 여부만 파악하였을 뿐 어떠한 유형의 파손인지는 파악이 되지 않는 어려움이 있다. 또한, 이를 판단할 수 있는 균일한 데이터 셋이 존재하지 않는다는 단점으로 인해서 제대로 된 유지 보수가 이뤄지지 않았다. 이러한 점을 개선하기 위해서 일본의 Omata 연구팀[3]은 최대 8가지 종류의 도로 파손을 탐지할 수 있게 하여 파손을 감지하여 운전자들이 촬영하여 업로드 시 촬영된 환경과 관계없이 해당 파손에 따른 필요한 유지보수를 파악하고 진행이 가능하게 해주는 모바일 애플리케이션 서비스를 개발하였다.

2.2 차량을 활용한 도로 파손 탐지

도로 파손 탐지와 관련된 서비스는 다양하다. 그러나 파손이 탐지된 도로를 정기적으로 지나치지 않는 이상 파손된 도로의 유지 보수 여부 및 상태 점검을 진행할 수 없다. 이에 따라 GPS와 같이 사전에 장착된 센서들이 존재하며, 정기적으로 특정 도로를 주행하는 버스와 같은 차량에 레이저 라인 스트리퍼(Laser Line Striper)와 카메라를 장착하여 차량에 부착한다. 다양한 차량 내에서 카메라와 레이저 라인 스트리퍼를 통해 수집된 데이터들을 사용하여 사람이 직접적으로 유지 보수하는 도로의 문제들을 차량들이 직접 도로 파손 정도를 탐지하고 도로 네트워크를 지속적으로 모니터링을 할 수 있는 시스템[4]을 개발하였다. 또한, [11]에서는 도로 표면 거칠기를 고려한 차량의 응답을 이용하여 도로의 손상을 탐지하는 방법을 연구하였다.

앞서 살펴보았던 기존의 관련 서비스와 시스템은 도로 파손 탐지는 물론 지속적인 모니터링이 가능하다. 그러나 주행을 멈추고 핸드폰으로 사진을 찍어야 탐지가 가능해지기 때문에 자동적으로 탐지가 가능하지 못하다는 점, 시스템이 GPS 등과 같은 모든 시스템이 장착된 차량에 부착이 되어야

한다는 점, 카메라를 통해서 측정값과 레이저 라인 스트리퍼의 측정값을 비교해야 한다는 점, 정기적으로 고정된 도로만 주행하는 차량에 부착이 되어야 한다는 점의 문제점이 있다.

그러므로 본 논문에서는 앞서 제시한 도로 파손 탐지를 주제로, RC카 (Radio Control Car) 시뮬레이터에 진동 센서와 GPS 센서를 부착된 아두이노 UNO를 탑재하여 시뮬레이터가 주행하면서 생성하는 진동 수치를 측정, 이를 통해 도로 파손 여부를 판단하는 시스템을 제안한다.

3. RC카 시뮬레이터 설계

본 장에서는 제안하는 시스템의 개요와 시스템 구성, 개발 환경에 대해 설명한다.

3.1 RC카 시뮬레이터 개요 및 기획

본 절에서는 제안하는 시스템의 전반적인 개요를 설명한다. 본 시스템은 도로 파손을 주제로 한 파손 감지 시스템이다. RC카 시뮬레이터로 주행하기 시작하면 진동 센서에서 차량이 주행하면서 도로 파손으로 인하여 증가되는 진동 수치들을 감지한다. GPS 센서의 경우 시뮬레이터가 야외에서 주행하면서 일정 주기로 측정된 위도와 경도 데이터를 저장한다. 시뮬레이터에 위치한 라즈베리 파이가 아두이노로부터 위의 센서 데이터들을 수합, 서버로 전달함으로서 외부 사용자가 차량의 자체 진동 기본 값 이상의 값들을 보고 파손 정도를 유추하는 시스템이다.

본 시스템은 다음과 같이 구성되어 있다. 주행은 아래 Fig. 1과 같은 RC카를 기반으로 제작된 시뮬레이터를 사용하여 진행된다. 시뮬레이터를 사용하여 측정 시 측정 순서, 측정된 시간, 측정된 센서 데이터 값, 그리고 측정된 위도와 경도 의 순서대로 Fig. 2와 같이 표로 구성되어 웹에서 실시간으로 확인할 수 있다. 이러한 테이블에 존재하는 데이터들 중에서 위도와 경도 값이 존재하는 데이터들의 경우 아래 Fig. 3과 같이 위도와 경도 자료를 사용하여 추정되는 장소들을 웹 구글 지도에 마커로 표시되는 서비스로 구성된다. 이 때, 지도에 표시되는 마커는 부연 설명으로 지도에 표시되는 값들은 측정된 시간, 측정된 센서 데이터 값 등을 담고 있다.

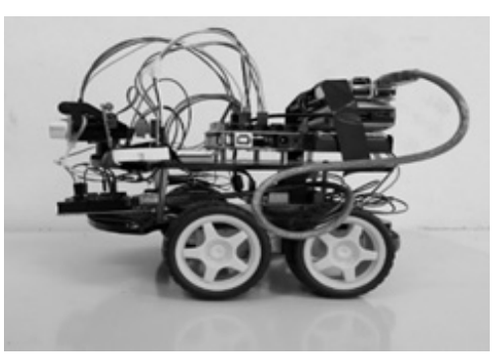

Fig. 1. The Proposed Prototype of RC Car Simulator

Roads in need of repair

Seq#	Time Stamp	Vib stat	Location
2	"2019-03-28T04:08:26.000Z"	6327	37.545950, 126.963627
2	"2019-03-28T04:08:28.000Z"	10453	37.545950, 126.963627
2	"2019-03-28T04:08:53.000Z"	6327	37.545950, 126.963627
2	"2019-03-28T04:09:00.000Z"	10453	37.545950, 126.963627
2	"2019-03-28T04:13:34.000Z"	7516	37.545950, 126.963627
2	"2019-03-28T04:13:35.000Z"	3368	37.545950, 126.963627
2	"2019-03-28T04:13:57.000Z"	15427	37.545950, 126.963627
2	"2019-03-28T04:14:09.000Z"	7516	37.545950, 126.963627
2	"2019-03-28T04:14:10.000Z"	3368	37.545950, 126.963627
2	"2019-03-28T04:14:35.000Z"	15427	37.545950, 126.963627
2	"2019-04-03T03:24:23.000Z"	5249	37.553819, 126.969544
2	"2019-04-03T03:24:53.000Z"	3319	37.545950, 126.963627
2	"2019-04-03T03:25:15.000Z"	3319	37.395037, 127.111124

Fig. 2. The Screen of Simulator Measurement Values

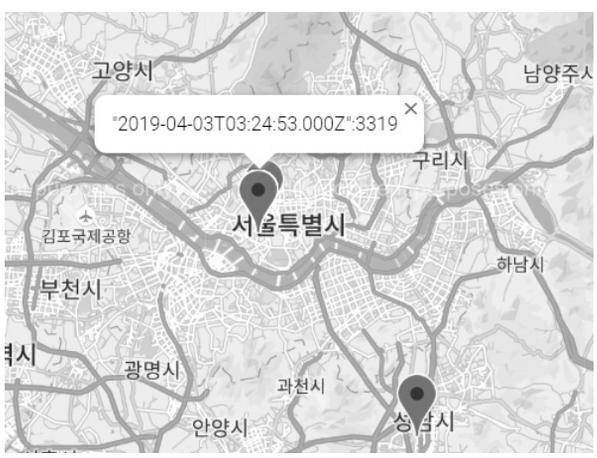

Fig. 3. The Screen of Simulator Measurement Location

본 논문에서 제안하는 시스템에서는 RC카 시뮬레이터를 사용하고 있다. RC카 시뮬레이터는 RC카, 아두이노 UNO, GPS 센서와 진동 센서, 라즈베리 파이 3 (이하 '라즈베리 파이')로 구성되어 있다. GPS 센서, 진동 센서는 센서 측정용 아두이노 UNO 와 직렬연결로 연결 되어 있으며, 해당 아두이노 UNO 와 라즈베리 파이 두 기기 사이에서 데이터를 주고받기 위해 사용한 통신은 시리얼 통신 (Serial Communication)을 이용하였다.

여기서 시리얼 통신은 데이터를 동일한 한 개의 통신선에 따라 비트 단위로 시간의 흐름에 맞춰 송수신하는 방법이다. 간단하고 비교적 통신 가능 거리가 길고, 제작비용이 저렴하여 비용 절감 효과가 크다는 장점이 있다[5]. 이러한 장점을 가진 통신을 사용함으로서 센서 와 라즈베리 파이는 USB 케이블을 통해서 안정적인 데이터 송수신이 가능함을 알 수 있다.

본 논문에서 제안하는 시뮬레이터를 기반으로 도로 파손 탐지 시스템의 구성도는 Fig. 4를 통해서 나타낸다.

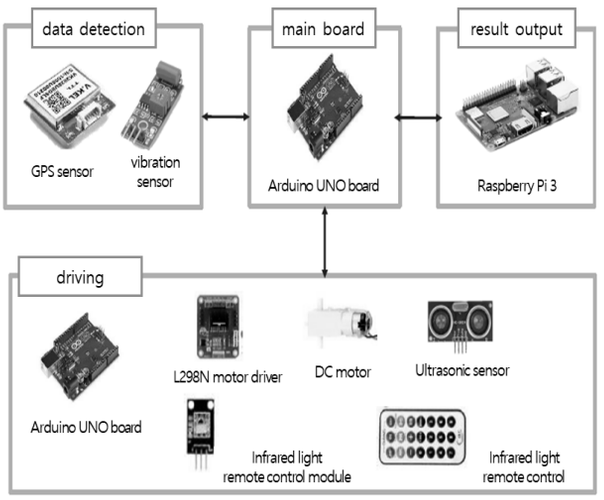

Fig. 4. The Road Damage Detection System Based on the Simulator

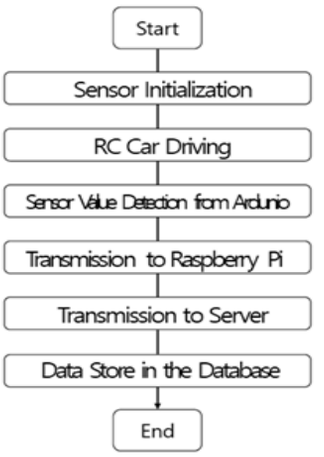

Fig. 5. The Flow Chart Based on Simulator

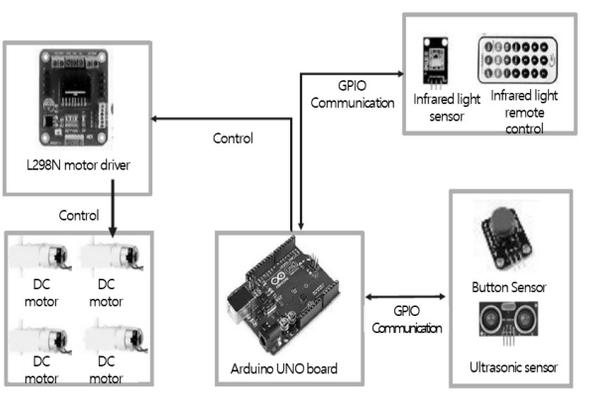

Fig. 6. The RC Car System

RC카는 'arduino.cc'사의 아두이노 UNO를 중심으로, 앞, 뒤, 오른쪽, 왼쪽 등으로 방향을 제어할 수 있도록 DC 모터를 4개를 사용하였으며, 이러한 DC 모터들을 제어해 줄 수 있는 L298N 모터 드라이버, 초음파 센서HC-SR04, 적외선 수신 센서 ky-022, 버튼 센서를 사용하였다.

주행 시 생성되는 진동 값을 측정하기 위해서 동일 회사의 아두이노 UNO와 진동 센서 SW-420, 주행 시 해당 위치를 얻기 위해서 GPS 센서 V.KEL-GPS를 사용하였다.

센서 데이터를 정제하고 웹 서버로 데이터를 수신하기 위해서 라즈베리 파이를 사용하였으며, 웹 서버로 수신된 센서 데이터들은 인터넷이 가능한 모든 원격지에서 사용자가 PC를 통해 모니터링 할 수 있도록 설계하였다.

3.2 RC카 시뮬레이터 시스템 구성

본 논문에서 제안하는 시스템은 RC카를 사용해서 주행을 하는 단계와 아두이노 UNO 에서 진동 센서와 GPS 센서들의 값을 연산을 통해 변환한 후 라즈베리 파이 에게 전달하는 단계, 라즈베리 파이에서 이러한 값들을 정제하여 서버로 전송해주는 단계, 라즈베리 파이에서 데이터베이스로 정보를 저장하는 단계들로 나눌 수 있다. 이러한 시스템 흐름도는 Fig. 5에서 확인할 수 있다.

이러한 흐름에 따라서 본 장의 3.2.1절에서는 RC카, 3.2.2절에서는 센서가 부착된 아두이노 UNO와 라즈베리 파이, 3.2.3절에서는 데이터를 저장하는 데이터베이스의 주요 기능에 대해서 설명한다.

1) RC카 구성

본 절에서는 시뮬레이터 내의 RC카가 아두이노와 모터 드라이버, DC 모터, 그리고 초음파 센서를 사용하여 주행하는 단계에 대한 구성을 설명한다. 아래 Fig. 6은 RC카의 구성도이다.

RC카의 핵심 기능은 주행과 방향 결정 두 부분으로 나눌 수 있다. 이에 따라 RC카의 기술은 크게 모터 부와 프로세서 부로 나눠진다. 모터 부는 전후에 각 2개씩 배치되어 총 4개의 DC 모터를 제어하는 역할을 하며, 모터 드라이버 L298N을 이용해 모터를 동작시킨다. 각 모터들은 전, 후 움직임을 제어하도록 하였으며, 좌, 우에 위치하는 DC 모터 간 속도를 조절함으로서 좌, 우 움직임도 제어할 수 있도록 하였다.

프로세서 부는 아두이노 UNO를 프로세서 처리로 사용하며, 컨트롤러에서 받은 명령어를 처리하여 모터 부를 제어하는 역할을 한다. 이 때, 프로세서 부에 전달받는 자동 주행과 수동 주행 두 가지 종류의 주행 모드가 있다. 이러한 주행 모드는 전환이 가능하다.

수동 주행 모드인 경우 적외선 수신 센서 Ky-022 센서와 적외선 리모컨을 사용하여서 리모컨 조작을 통해서 RC카를 주행할 수 있다.

자동 주행 모드인 경우 직렬로 연결된 초음파 센서 HC-SR04를 사용하여 RC카 앞에 존재하는 장애물과의 거리를 탐지 후 해당 거리에 따라서 주행을 자동적으로 조절할 수 있다. 다음 Fig. 7은 구현이 완성된 RC카의 모습의 예시를 나타낸다.

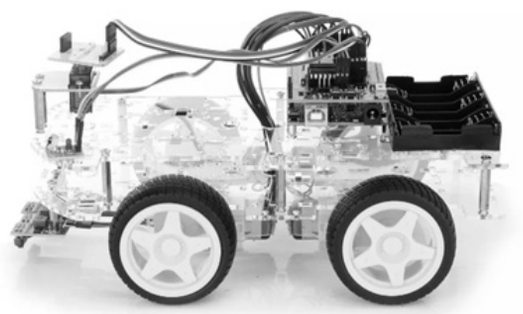

Fig. 7. An Prototype of Proposed RC Car

2) 센서 보드 구성

본 절에서는 아두이노 UNO에서 센서들의 값을 연산을 통해 변환한 후 라즈베리 파이에게 전달, 라즈베리 파이에서 센서들의 측정값을 서버에게 전달하는 단계에 대한 구성을 설명한다. 아래 Fig. 8은 해당 센서 보드의 구성도이다.

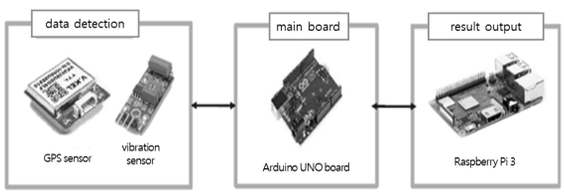

Fig. 8. The Sensor Boards of System

센서 보드에는 크게 아두이노 UNO 와 진동 센서 SW-420, GPS 센서 V.KEL, 라즈베리 파이로 구성되어 있다. 센서 보드에서는 센서로부터 데이터를 전달 받는 기능과 라즈베리 파이에서 서버로 데이터를 전달하는 기능 두 가지로 나눌 수 있다. 이에 따라, 센서 보드에서는 크게 데이터 처리 부와 데이터 전달 부로 나눌 수 있다.

데이터 처리 부는 아두이노 UNO 를 사용하며, 진동 센서와 GPS 센서로부터 데이터를 처리하는 역할을 한다.

데이터 전달 부는 라즈베리 파이를 사용하며, 시리얼 통신을 통해서 아두이노 UNO 로부터 전달받은 데이터를 분류하여 서버에게 전달하는 역할을 한다.

3) 데이터베이스 구성

본 절에서는 센서 에서 생성되는 모든 데이터들을 저장하는 데이터베이스의 스키마 및 테이블 구성에 대해서 설명한다. 데이터베이스는 크게 라즈베리 파이에서 수신한 데이터를 저장하는 log 테이블과 서버에서 웹에 디스플레이 하는 데이터를 저장하는 results 테이블로 구성되어 있다. log테이블과 results 테이블은 저장되는 데이터들의 종류가 다를 뿐 기본적인 스키마는 동일한 서로 독립된 테이블이다. 이 아래 Fig. 9를 통해 이와 관련된 두 테이블의 스키마와 애트리뷰트에 대해서 설명이 가능하다.

먼저 측정되는 데이터들의 수를 알려주는 id는 자동적으로 증가하는 auto_increment 특성과 함께 PRIMARY key이며, seq는 측정 횟수를 측정해주는 키이다. 측정 기기의 이름이나 고유 식별 번호와 같은 데이터를 저장하는 device, GPS 센서의 값이 측정되었을 경우 측정된 해당 위도, 경도의 값이 저장되는 location, 진동 센서의 측정값이 저장되는 value, 데이터를 수신하는 기기의 인터넷규약주소 (Internet Protocol, IP) 주소가 저장되는 ip와 측정 시간이 저장되는 time 등이 있다.

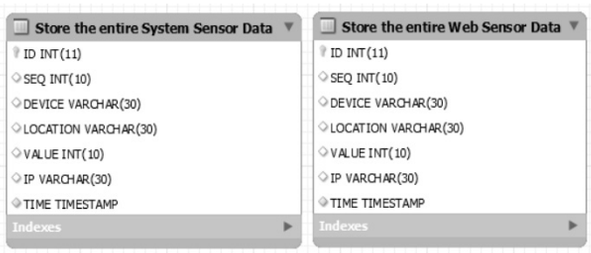

Fig. 9. The Schema of Database Table

3.3 RC카 시뮬레이터 시스템 개발 환경

본 논문에서 제안하는 시스템의 개발환경은 Table 1과 같다.

Table 1. Experimental Environments

OS	Linux (Raspbian)
Development Language	Python(Raspberry Pi), C++ (Arduino), Node.js (Server), SQL (Database)
Server	Web Application Server

1) 운영 체제

해당 시스템은 라즈베리 파이에 기본적으로 내제되어 있는 라즈비안 (Raspbian) 이라는 리눅스 운영체제를 사용하였다. 리눅스는 다중 사용자, 다중 작업(멀티태스킹), 다중 스레드를 지원한다는 장점을 가지고 있다. 또한, 네트워크를 이용하는 컴퓨팅에서 사용이 용이하다[6].

2) 아두이노

아두이노는 하드웨어와 소프트웨어 모두를 기반으로 하는 오픈 소스 전자 플랫폼이다. 아두이노의 하드웨어는 센서와의 연결을 통해서 다양한 기능 구현이 가능한 반면, 소프트웨어의 경우 내장된 마이크로 컨트롤러와 아두이노 고유의 소프트웨어를 사용하여 부착한 센서들을 제어하는 함수들을 구현할 수 있다. 아두이노 소프트웨어는 다양한 운영체제에서 실행이 가능하기 때문에 이점이 있다[7].

3) 개발 언어

아두이노 UNO 에 기본적으로 내제되어 있는 C++ 언어를 사용하여 주행 및 센서 데이터 처리 기능을 개발하였다. 라즈베리 파이에 기본적으로 내제되어 있는 Python 3.5을 사용

하여 아두이노 UNO 로부터 데이터 송신 및 서버로의 수신 기능을 개발하였으며, Node.js를 설치하여 웹 서버 구축 및 라즈베리 파이에서 전달받는 데이터들을 웹에 디스플레이 하는 기능을 개발하였다. 또, MySQL을 설치하여 데이터 값을 저장하는 기능을 개발하였다.

4) 웹서버

데이터 수신과 송신이 동시에 일어나는 라즈베리 파이에 큰 부담이 가지 않는 HTTP 통신이 가장 적합하다고 판단하였고, HTTP 통신을 위한 웹 애플리케이션 서버를 구축하였다.

4. 시스템 구현

본 장에서는 3장에서 다뤄진 제안하는 시스템의 기능 구현에 대해서 각 절에 해당하는 기능들의 구현에 따른 알고리즘 및 기능들을 설명한다.

4.1 RC카 시뮬레이터 시스템 설계

Fig. 10은 본 논문서 제안하는 시스템의 전체적인 기능에 대한 설계이다. 전원이 들어온 후 센서 초기화 과정을 거친 시뮬레이터는 먼저 수동으로 주행을 진행한다. 주행 중 버튼 모듈의 입력이 홀수 번으로 들어왔을 때, 시뮬레이터는 자동 주행 모드로 주행 방법을 바꿔 자율적으로 주행을 시작한다. 이때, 초음파 센서 HC-SR04를 사용하여 장애물 탐지 및 회피 알고리즘을 진행하게 된다. 반면 버튼 모듈의 입력이 전혀 들어오지 않았거나 짝수 번으로 입력을 받았을 경우 기존에 진행하던 수동 주행을 진행하게 된다.

주행을 진행함과 동시에 센서 에서도 진동 센서와 GPS 센서에서 자동적으로 측정된 수치 데이터를 입력 받게 되고, 이를 소켓 통신을 통해서 라즈베리 파이로 전달하게 된다. 이러한 상황을 진행하면서 라즈베리 파이에서는 센서 데이터 처리 알고리즘이 활용된다. 더 나아가, 앞서 명시하였듯이 소켓 통신은 인터넷으로 진행되지 않고 USB 유선 케이블로 진행되기 때문에 인터넷의 존재 여부는 관계가 없다.

그러나. 라즈베리 파이와 서버간의 HTTP 통신의 경우 인터넷을 필요로 한다. 이로 인해 라즈베리 파이 내에서는 먼저 서버와의 연결 여부를 묻게 되고, 연결이 불가한 상황의 경우 라즈베리 파이 내에 설계된 자체 데이터베이스 테이블에 저장하게 된다. 반면 서버와의 연결이 원활한 경우, 라즈베리 파이에서는 HTTP 통신을 통해서 서버에 처리한 데이터를 전달하게 된다. 서버와의 네트워크 연결 상황과 관계없이 데이터베이스에 저장이 될 때, 센서 데이터 분류 알고리즘이 활용된다.

작업이 끝난 라즈베리 파이는 RC카 시뮬레이터의 전원이 아직 들어온 상태인지 파악하게 되고, 전원이 들어와 있는 경우는 진행 중인 주행 모드를 따라가게 된다. 반면, 전원이 더 이상 공급되지 않아 그 이상의 주행이 어려운 경우 시스템을 종료하게 된다.

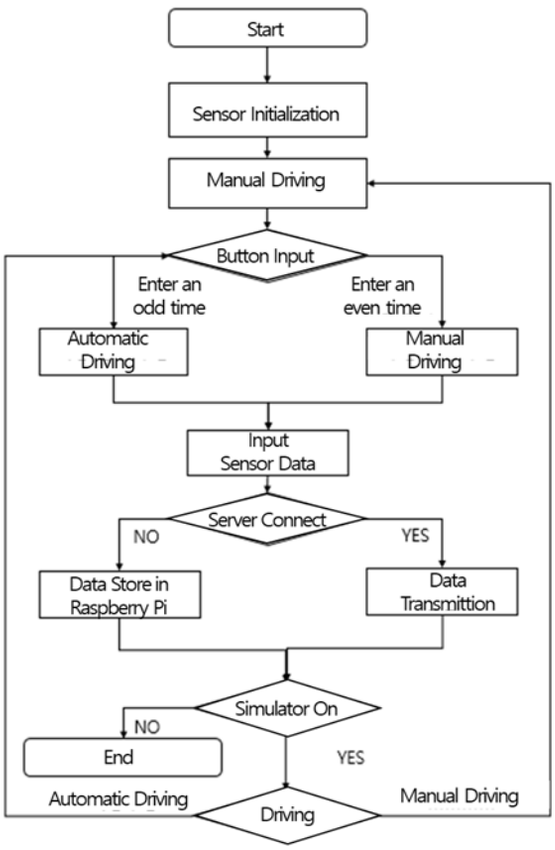

Fig. 10. The Progress Chart of Overall System

4.2 RC카 시뮬레이터 응용 및 구현

1) RC카

본 절에서는 3.2.1절에서 언급된 RC카의 수동 주행과 자동 주행에 관련된 알고리즘을 설명한다. Fig. 11과 Fig. 12는 각각 RC카의 수동 주행 작동 알고리즘과 RC카의 자동 주행 작동 알고리즘에 대해서 설명한다.

Fig. 11이 표현하는 RC카의 수동 주행 알고리즘은 다음과 같은 순서로 진행된다. 먼저 RC카에 전원이 들어오면 데이터를 전달해주는 적외선 센서나 모터 드라이버 등이 초기화된다. 센서 초기화 후에는 적외선 센서로부터 적외선 리모컨에서 전달한 제어 명령어들을 수신하고, 전달받은 명령에 따라서 속도의 증감이나 방향의 제어 작업을 수행하도록 하였다. 이러한 알고리즘의 결과로 RC카는 주행 시 사용자의 방향 및 속도 제어를 통해서 장애물과 부딪히지 않고 원활한 주행이 가능하게 된다. 다음 Fig. 12는 이러한 수동 주행이 구현되는 모습을 사진으로 표현하는 것이다.

Fig. 13에서 표현하는 RC카의 자동 주행 알고리즘은 다음과 같은 순서로 진행된다. 먼저 RC카에 전원이 들어오면 데이터를 전달해주는 초음파 센서나 모터 드라이버 등이 초기화된다. 센서 초기화 후, RC카는 초음파센서 로부터 수신되는 주행 도로 앞에 존재하는 장애물과의 거리를 탐지하면서 주행을 시작한다.

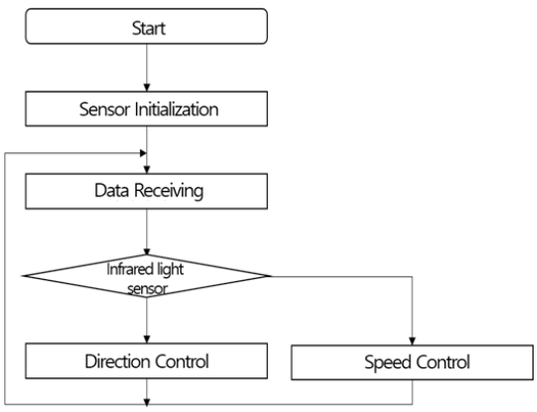

Fig. 11. The Flow Chart of Manual Driving using RC Car Simulator

Fig. 12. The Manual Driving using RC Car Simulator

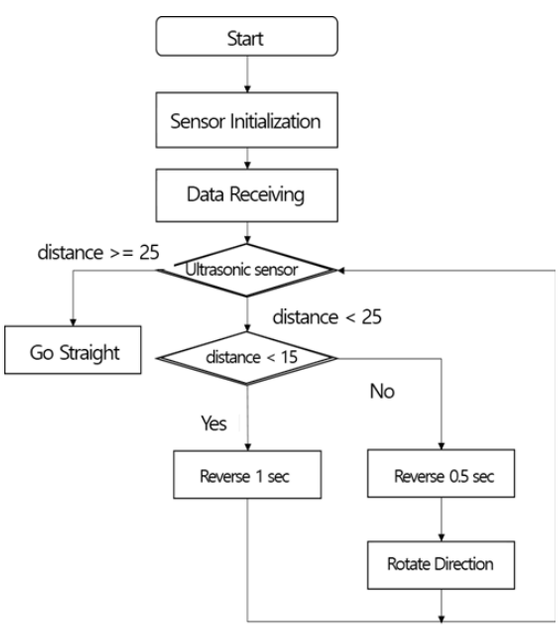

Fig. 13. The Flow Chart of Automatic Driving using RC Car Simulator

차량 앞에 존재하는 장애물과의 거리가 RC카 시뮬레이터의 세로 길이인 25cm 보다 클 경우에는 정상적으로 직진을 한다. 그러나 25cm 이내인 동안 차량이 방향 변경을 위해서 90도로 회전할 경우 RC카 시뮬레이터의 가로의 길이인 15cm 이내로 접근하였을 때 차량은 회전 시 장애물과의 충돌을 방지하기 위해 1초간 후진을 한 후, 초음파 센서를 사용하여 거리를 측정한다.

반면 25cm 이내이되 15cm보다 먼 거리라고 측정된 경우 RC카는 방향 전환을 위해서 회전 시 전방에 위치한 장애물과의 충돌을 방지하기 위해 0.5초간 후진을 한 후, 무작위로 오른쪽이나 왼쪽으로 90도 회전을 하여 방향을 전환한 다음 다시 초음파센서를 사용하여 장애물과의 거리를 측정한다. 이러한 알고리즘의 결과로 RC카는 주행 시 사용자의 개입 없이도 장애물과 부딪히지 않고 원활한 주행이 가능하게 된다. 다음 Fig. 14는 이러한 자동 주행이 구현되는 RC카 시뮬레이터를 사진으로 표현한다.

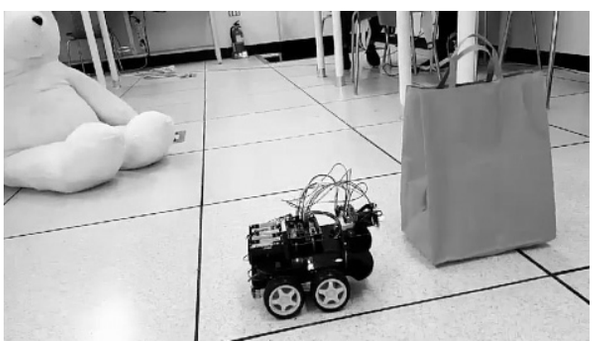

Fig. 14. The Automatic Driving using RC Car Simulator

2) 센서

본 절에서는 3.2.2절에서 언급된 센서 에서 데이터 처리부의 센서 데이터 처리 알고리즘과 데이터 전달 부에서 데이터 분류에 사용하는 데이터 분류 알고리즘에 대해서 설명한다. 아래 Fig. 15와 Fig. 16, Fig. 17은 센서 데이터 처리 알고리즘을, Fig. 18은 데이터 분류 알고리즘에 대해서 설명한다.

Fig. 15의 센서 데이터 처리 알고리즘은 다음과 같은 순서로 진행된다. 먼저 아두이노 UNO 에 전원이 들어오면 데이터를 전달해주는 진동 센서와 초기화된다. 진동 센서의 경우 아두이노 UNO 에 전원이 들어옴과 동시에 자동적으로 데이터 값을 수신하기 하는 반면 GPS 센서의 경우 건물이나 터널과 같이 실내에 있을 경우 근처 기지국과의 통신이 저하되어 현재의 정확한 위치를 불러올 수 없다.

이에 따라 GPS 센서의 값이 수신 될 수 없는 경우는 GPS 데이터 는 'IN'이라는 문자열 값을 가지게 되며, 수신이 가능한 경우는 GPS 데이터 는 현재 위치의 위도와 경도가 된다. 이러한 데이터들은 3.1절에서 언급한 시리얼 통신을 통하여 라즈베리 파이에 전송된다.

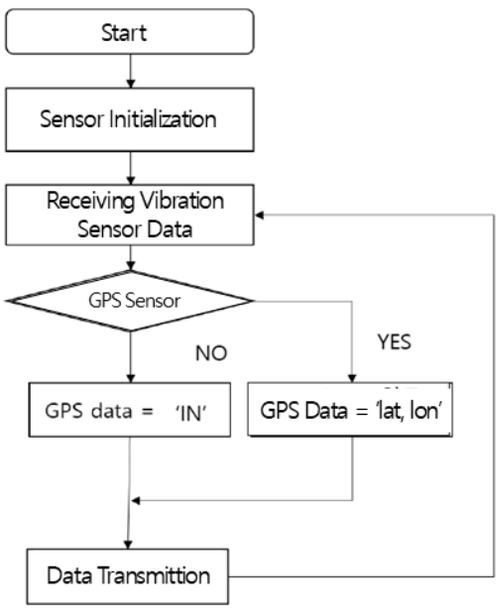

Fig. 15. The Flow Chart of Processing Sensor Data

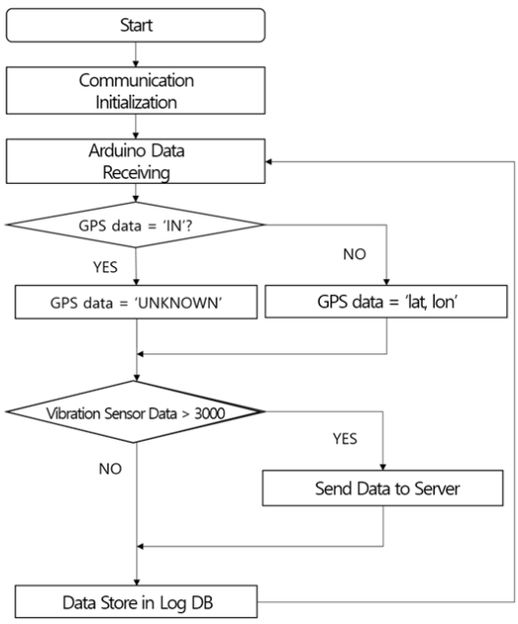

Fig. 16. The Flow Chart of Sensor Data Classification

Fig. 16의 알고리즘은 다음과 같은 순서로 진행된다. 먼저 라즈베리 파이에 전원이 들어오면 라즈베리 파이는 아두이노 UNO 와의 시리얼 통신을 초기화한다.

아두이노 UNO로부터 시리얼 통신으로 전달받는 데이터들을 받아 먼저 GPS 센서 값이 있는GPS 데이터를 확인하게 된다. 시리얼 통신을 통하여 데이터를 전달받은 라즈베리 파이에서 데이터베이스로 정보를 저장할 때 사용된 알고리즘에 대해서 설명한다.

Fig. 15의 알고리즘에서 설명한 바와 같이 기지국과의 통신에 실패하여 'IN'이라고 저장되어 있는 경우 구체적인 위치를 판단할 수 없어 첫번째 값이 IN일 경우 실내에서 측정된 값이라 판단, 진동 센서의 측정 값을 읽어 들이고 위치 정보는 실내 라 판단, 구체적인 위도, 경도를 알 수 없기에 "UNKNOWN"이라 지정하여 전달한다.

반면 그렇지 않은 경우 진동 측정값과 GPS 센서로부터 위치 데이터가 전달되었을 경우 위치 데이터를 전달받는다.

세 번째, 진동 센서의 측정값이 차체에서 발생되는 평균 진동 값인 3000 이상인지를 따져본다. 측정된 진동 값이 평균값을 초과하는 경우 주행한 도로가 점검이 필요한 상태라고 판단, 측정된 값을 HTTP 통신을 통해서 서버로 전달해준다.

전체 측정된 데이터에 대한 로그 파일인 log2.txt와 측정된 모든 데이터를 저장하는 데이터베이스 log 테이블에 저장한다. 그렇지 않을 경우 log데이터베이스에만 저장하도록 한다.

3) 서버 데이터 디스플레이

본 절에서는 3.2.4 절에서 언급된 서버에서 데이터 저장 및 웹에서 시각화하는 기능을 설명한다. Fig. 17에서 언급되는 알고리즘을 통해서 순서대로 웹에서 데이터베이스에 데이터 저장 순서를 파악할 수 있다. 먼저, 라즈베리 파이에서 전달받은 정보가 담긴 쿼리 변수를 초기화한다. 다음으로 IF 구문을 사용하여 쿼리 변수가 라즈베리 파이에서 제대로 된 데이터를 받아 생성된 쿼리일 경우, 데이터베이스에 연결하여 쿼리에 대해서 삽입 기능을 실시한다. 그런 다음 쿼리를 JSON 유형으로 변환하여 라즈베리 파이 내부에 기록을 저장하는 로그 파일에 저장한 후, 삽입이 끝났다는 콘솔창 메시지를 출력한다. 만일, 라즈베리 파이에서 데이터를 제대로 받지 못하여 에러가 발생한 경우, 에러의 내용을 콘솔에 출력한다다.

Fig. 18을 통해서는 웹에서 데이터베이스에 저장된 값들을 조건문을 통하여 불러와 테이블로 보여줄 수 있다. 먼저 데이터베이스로부터 전달받은 쿼리들을 'info'라는 이름의 변수로 지정한다. 해당하는 쿼리에 대한 결과가 존재할 경우, 데이터베이스에 연결하여 선택된 쿼리들을 전달 받는다. 그 후, 측정 순서를 나타내는 'id', 측정 시간대를 나타내는 'time', 센서의 측정값을 나타내는 'value', 측정 위치를 나타내는 'location'에 대한 값들을 문자열 형으로 변환한다. 그런 다음, 변환된 문자열들을 테이블에 값으로 집어넣어 준 후 테이블을 반환한다. 이때, 쿼리에 해당하는 결과가 하나도 존재하지 않을 경우 콘솔 출력창으로 발생한 에러를 출력해준다.

더 나아가 웹에서 GPS 데이터가 존재하는 데이터들은 Google Maps를 사용하여 지도에 표시해줄 수 있다. 먼저, 데이터베이스에서 가져올 쿼리를 place, 쿼리 내에서 측정 시간에 해당하는 'time' 애트리뷰트, 지도에 표기할 marker라는 변수에 (37, 127.1)에 해당하는 값들을 지정하는 초기화 과정을 거친다. 이때, 만약 측정 일자가 현재로부터 2주내의 데이터일 경우 데이터베이스에 연결하여 쿼리문을 통해 결과를 가져온다. 이후, 선출된 쿼리 내에서 측정 위치를 나

타내는 'location', 센서 측정값을 나타내는 'sensor data', 측정 시간을 나타내는 'time'에서 결과 값을 도출한다. 도출된 결과 값들을 전부다 문자열 형으로 변환시켜준 뒤, Google Maps에 문자열로 형변환이 일어난 location 데이터에 해당하는 위치를 찾는다. 해당하는 위치를 찾은 경우, marker를 사용하여 해당하는 위치에 마커로 표시를 해준다.

[Algorithm 1] Load to Database

Input: information from Raspberry Pi
Output: Console.log
Algorithm:
1: **Initialize:** query = information from Raspberry Pi
2: **IF** (query != err) **THEN**
3: connect to database
4: insert query to the database
5: convert query into JSON format
6: insert JSON to log.txt file in raspberry pi
7: return console.log("Injection is done!")
8: **ELSE**
9: return console.log(err)
10: **ENDCASE**

Fig. 17. Load to Database Algorithm

[Algorithm 2] Display in Web

Input: queries from Database
Output: Table
Algorithm:
1: **Initialize:** info = queries from Database
2: **IF** (info != null) **THEN**
3: connect to database
4: select query to the database
5: change id, time, value, location attribute data of query into string
6: insert string data to table
7: return table
8: **ELSE**
9: return console.log(err)
10: **ENDCASE**

Fig. 18. Display in Web Algorithm

5. 결 론

본 논문에서는 RC카 시뮬레이터를 기반으로 주행을 하면서 도로 파손 여부를 진동 센서의 수치 데이터로 유추, 의미가 있다고 판단되는 값을 서버로 전달하여 웹에서 표와 지도로 시각화하여 보여주는 시스템을 제작하였다. 최근 다양한 원인으로 발생되는 포트홀과 같은 도로 파손 문제들은 국내뿐만 아니라 국제적으로 심각한 문제이다. 본 논문에서 제안하는 시스템은 차량이 주행을 하면서 실시간으로 발생되는 진동 센서의 수치로 도로 파손 여부를 판단할 수 있게 하여 카메라와 같이 빛과 같은 환경적 요인에 영향을 받지 않으며, GPS 센서를 통해서 위치 정보를 저장하여 웹에 디스플레이를 통해서 구체적인 위치를 파악할 수 있다. 그렇기 때문에 파손 도로 촬영과 같이 부가적인 행동이나 서비스를 요구하지 않으며 안정적으로 데이터를 얻을 수 잇다는 긍정적인 영향을 미칠 수 있을 것이다.

향후 목표는 이렇게 설계한 시스템을 실제로 활용하여 실제 도로의 파손 상태를 분석하는 실험을 진행하고자 한다. 집산 된 데이터를 통해서 국내 도로들 간의 파손 주기나 평균적인 진동 값을 얻음으로써 이를 통해 도로의 전반적인 파손 여부 확인 및 보수 작업이 시행된 도로에 대한 지속적인 모니터링이 가능하도록 유도하는 것이다. 더 나아가 시각 장애를 앓는 보행자들이 사용하는 보도 블록과 같이 일정한 굴곡이 존재해야 하는 도로의 파손을 탐지할 수 있으며, 휠체어나 보행보조기와 같이 바퀴가 달린 보조 기구를 사용해야 하는 거동이 불편한 사람들을 위한 보도 블록 및 도로 탐지 서비스 등으로 확장되어 구현될 수 있을 것이다.

References

[1] 박성훈, "전국 곳곳에 '싱크홀 공포'...왜", 문화일보, [Internet]. Available: http://www.munhwa.com/news/view.html?no=2018091201031027109001

[2] D. Kim, J. Jeon, and D. Kim, "Predictive Modeling of Pavement Damage Using Machine Learning and Big Data Processing," *Journal of Korean Society of Hazard Mitigation*, Vol.19, No.1, pp.95-107, 2019.

[3] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, and H. Omata, "Road Damage Detection Using Deep Neural Networks with Images Captured Through a Smartphone," *Computer-Aided Civil and Infrastructure Engineering*, Vol.33, No.12, pp.1127-1141, Jun. 2018.

[4] C. Mertz, "Continuous Road Damage Detection Using Regular Service Vehicles," in *Proceeding of the 18th ITS World Congress*, Orlando, 2011.

[5] J. Kim, "A Study on the Efficient Protocal Structure of Monitoring System Using Serial Communication," M.S. dissertation, Sungkyunkwan University, Seoul, 2015.

[6] H. Yang, B. Kim, and D. Kim, "Implementation of Raspberry Pi Rhythm Game using UART Communication in the different Operating Systems," *Journal of Digital Contents*

Society, Vol.20, No.3, pp.647-655, Mar. 2019.
[7] "What is Arduino?", [Internet]. Available: https://www.arduino.cc/en/Guide/Introduction
[8] MOLIT STATISTICS SYSTEM, URL: stat.molit.go.kr
[9] P. Li, H. Xu, and B. Song, "A Novel Method for Urban Road Damage Detection Using Very High Resolution Satellite Imagery and Road Map," *Photogrammetric Engineering & Remote Sensing*, Vol.77, No.10, pp.1057-1066, Oct. 2011.
[10] L. Gong, L. An, M. Liu, and J. Zhang, "Road Damage Detection from High-Resolution RS Image," in *Proceeding of the 2012 IEEE International Geoscience and Remote Sensing Symposium*, Munich, 2012.
[11] Z. Li and F. T. K. Au, "Damage Detection of Bridges Using Response of Vehicle Considering Road Surface Roughness," *International Journal of Structural Stability and Dynamics*, Vol.15, No.3, 2015.

이 유 나
https://orcid.org/0000-0001-9339-6451
e-mail : ylee5813@korea.ac.kr
2019년 숙명여자대학교 IT공학과(이학사)
2019년 ~ 현 재 고려대학교 컴퓨터학과 석사과정
관심분야 : RC카, 데이터베이스, 머신러닝, 빅데이터, 컴퓨터 비전, 딥러닝, 최적화(Optimization)

박 영 호
https://orcid.org/0000-0002-5284-9589
e-mail : yhpark@sm.ac.kr
1992년 동국대학교 컴퓨터공학과 (공학석사)
2005년 한국과학기술원 전산학과(공학박사)
1993년 ~ 1999년 한국전자통신연구원 교환전송연구단 선임연구원
2005년 ~ 2006년 한국과학기술원 첨단정보기술연구센터 연구원
2005년 ~ 2006년 동국대학교 컴퓨터멀티미디어학과 겸임교수
2006년 ~ 현 재 숙명여자대학교 IT공학과 교수
관심분야 : 데이터 분석(Data Analytics), 정보검색(Information Retrieval), 감성공학(Emotional Computing), 기계학습(Machine Learning), 데이터베이스 관리 시스템(DBMS), IT 융합(IT Convergence), XML

임 선 영
https://orcid.org/0000-0002-2529-2826
e-mail : sunnyihm@sm.ac.kr
2011년 숙명여자대학교 멀티미디어과학과 (이학사)
2013년 숙명여자대학교 멀티미디어과학과 (이학석사)
2017년 숙명여자대학교 IT공학과(공학박사)
2017년 ~ 2018년 숙명여자대학교 빅데이터활용 연구센터 책임연구원
2018년 ~ 2019년 숙명여자대학교 IT공학과 초빙교수
2019년 ~ 현 재 숙명여자대학교 빅데이터활용 연구센터 책임연구원
관심분야 : 데이터 분석(Data Analytics), 데이터베이스, 정보검색(Information Retrieval), Top-k 질의처리, 서브시퀀스 매칭, 머신러닝, 빅데이터