

영상의 분산 처리를 통한 무인 주행 시스템



이름	학번	이메일	전화번호
정세진	200911418	jsjj0728@gmail.com	010-5107-7822
김진현	200911381	zechekei@gmail.com	010-4440-4718
홍호탁	201114195	smileshot@naver.com	010-4594-7581

지도 교수 : 장천현 교수님 (인)

Index

1. 개발 동기
2. 관련 연구
 - 2.1. 무인 자동차
 - 2.2. 라즈베리 파이
 - 2.3. 영상 처리
 - 2.4. 분산 처리
3. 영상의 분산 처리를 통한 무인 주행 시스템
 - 3.1. 차량
 - 3.2. 서버
 - 3.3. 단말
4. 차량 하드웨어 구현
5. 특징 및 활용 방안
6. 역할 분담
7. 일정 계획
8. 참고 자료

1. 개발 동기

무인자동차 연구의 시발점은 DARPA에서 자체적으로 무인자동차 시스템을 개발하고 있었으나 별다른 진전이 없어, 민간기업을 끌어들이기 위해 Urban Challenge 무인자동차 경주대회를 개최하자 참여가 급증하였고, 기술 개발이 활발해졌다. Urban Challenge에서 우승자들이 현재 구글에서 제작한 구글카가로 무인자동차의 전망이 더욱 커졌다.

무인자동차에는 현재의 위치를 cm수준으로 정밀하게 파악하는 기술로 GPS, 레이저 스캐너, 레이더, 카메라는 기본이고, 수많은 첨단센서가 부착된다. 이렇게 수많은 센서가 부착됨으로 무인자동차를 제작하는 비용이 상당히 많이 들어갈 것이다. 여러 개의 센서의 역할을 한 개의 카메라가 대신하게 되면 무인자동차를 제작하는 비용은 절감하게 될 것이다.

본 프로젝트에서는 소형 무인자동차와 이를 제어하는 시스템을 제작한다. 보통 무인자동차에 부착되는 센서를 제외하고 카메라만으로 무인주행이 얼마나 가능한지 확인해보기로 한다. 소형 무인자동차는 저전력, 저사양 초소형 PC인 RaspberryPI를 사용하여 하드웨어를 구성한다. RaspberryPI 카메라 모듈 기반의 영상처리를 통한 무인주행을 보일 것이다. 또한 저사양 PC인 만큼 영상을 처리할 때 cpu과부하를 분산처리 시스템을 도입하여 이를 해결할 것이다.

2. 관련 연구

2.1 무인자동차

감속 및 가속, 조향 등의 명령을 내리는 중앙제어 장치의 명령에 따라 필요한 작동을 취하는 액추에이터 등으로 구성되며, 운전자의 조작 없이 자동차 스스로 주행환경을 인식하여 목표지점까지 운행할 수 있는 자동차이다

2.1.2 핵심 기술

무인자동차 시스템 및 그를 위한 Actual System, 비전 및 센서를 이용한 시가정보의 입력 및 처리장치, 무인화 운행에 적합한 조향 알고리즘, 통합관제 시스템과 운행감시 고장진단 체제, 지능제어 및 지능운행 장치 등이 필요하다.

2.1.3 관련 프로젝트

관련 프로젝트에는 CyCab, CyberCab, MobiVIP, CityMobil등이 있으며 CyCab 프로젝트는 미래형 자동차 프로젝트의 첫 번째 프로토타입으로 1996년 개발된 프로젝트로서 보행자 사이를 안전하게 돌아다닐 수 있는 자동차와 비슷한 로봇을 개발한다. CyberCab 프로젝트는 INRIA 와 Yamaha가 손을 잡고진행한 프로젝트로서 플로리다에서 개최된 2002년 엑스포에서 25대 CyberCab 들이 채택되어 실제로 운영을 하였는데, 대회장의 4m 높이에 있는 전망대까지 사람들을 운반하는 등 최대 4명이 탈 수 있는 시스템으로 무인자동차 시스템이 상용화할 수 있다는 가능성을 보여줬다. MobiVIP 프로젝트는 2004년에 시작한 프로젝트로서 저속의 반개방형 자동운전차량의 도시지역 적용을 목표로 GPS 시스템을 이용하여 모든 차량의 움직임을 추적하고 CyCab 차량에 전후방카메라를 설치하고, GPS와 gyrometer/odometer등 관성센서를 활용 하였다. City Mobil 프로젝트는 최초로 상용화를 전제로 시작했다는데 큰 의의가 있는 프로젝트이다. 현재는 구글 무인자동차 프로젝트에 많은 관심이 집중되고 있다.



그림 2

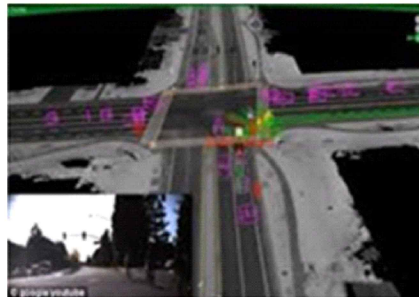


그림 3

2.1.4 국내 무인자동차 종류

2009년 KIST에서 개발한 가로수가 많은 환경에서도 스스로 도로를 달릴 수 있는 KUVE, ETRI에서 개발한 자율주행 ESTRO 전기자동차, GM에서 개발한 운전자가 모니터 상에 목적지를 지정하면 운전하지 않아도 스스로 목적지까지 주행이 가능한 BOSS등이 있다.

2.2 라즈베리 파이

라즈베리 파이는 아이들이 저렴한 가격으로 쉽게 접할 수 있는 교육용 컴퓨터를 목표로 개발되었다. 다양한 OS를 설치하고 설정을 변경하거나 필요한 프로그램을 설치하고, 또는 프로그래밍을 공부하는 등 자유롭게 컴퓨터를 가지고 놀며 배울 수 있도록 만들어졌다. 라즈베리 파이를 이용하여 실제 게임기에 연결할 수 있는 게임패드나 정해진 시간마다 화분에 물을 뿌려주는 간이 스프링클러도 제작할 수 있는 등 다양한 형태의 창작물들을 손쉽게 만들어 낼 수 있다. 라즈베리 파이는 거의 모든 분야에서 활용될 수 있는 무한한 잠재력을 가졌다.

2.2.1 라즈베리 파이 종류

라즈베리 파이에는 모델 a, 모델 b 이렇게 두 가지가 있다. 메모리 용량이 모델 a가 256MB 모델 b가 512MB로 모델 b의 메모리 용량이 더 많다. 이더넷도 모델 a는 없지만 모델 b는 10/100 이더넷 RJ45가 있다.

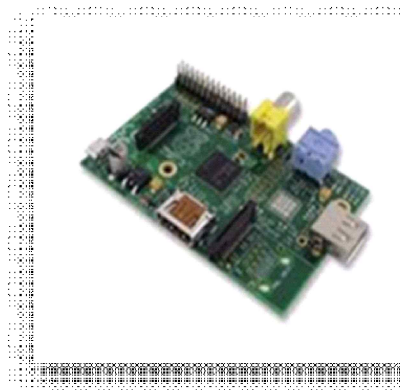


그림 4

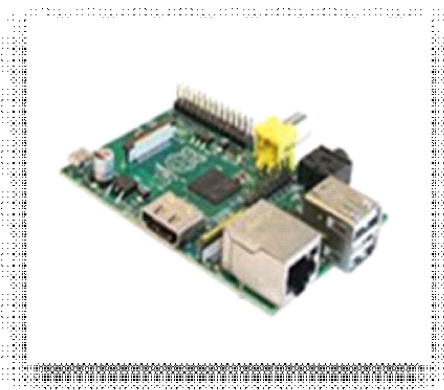


그림 5

2.2.2 모터제어

모터의 방향제어와 속도제어가 가능한 모터 구동 드라이브를 구매하거나, SN754410NE, L293D 등 IC칩을 구매하여 직접 회로를 구성해 모터제어를 하는 방법이 있다.

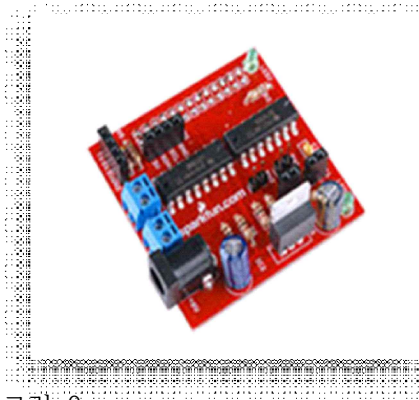


그림 6



그림 7

2.3 영상처리

2.3.1 차선 인식

무인 자동차가 도로를 주행할 때 영상정보의 획득은 필수적이므로 도로에서 안전하게 주행하기 위해서는 취득한 영상으로부터 무인 자동차가 주행하는 차선과 정지선을 인식해야 한다. 이러한 목적을 달성하기 위해 HSI컬러 모델에서 관심영역을 지정하여 빛의 강도의 분포차이와 fuzzy c-means algorithm을 사용하여 차선을 검출하는 알고리즘, CIELab 컬러 모델에서 색상 클러스터링을 통하여 차선을 검출하는 알고리즘, 기본적인 차선의 색상정보를 기반으로 하여 동일한 색상, 차선의 모양, 크기, 움직임 정보 등을 이용하여 실제 차선을 검출하는 알고리즘 등 색상정보와 차선의 형태적 특성을 사용하는 다양한 알고리즘이 있다.

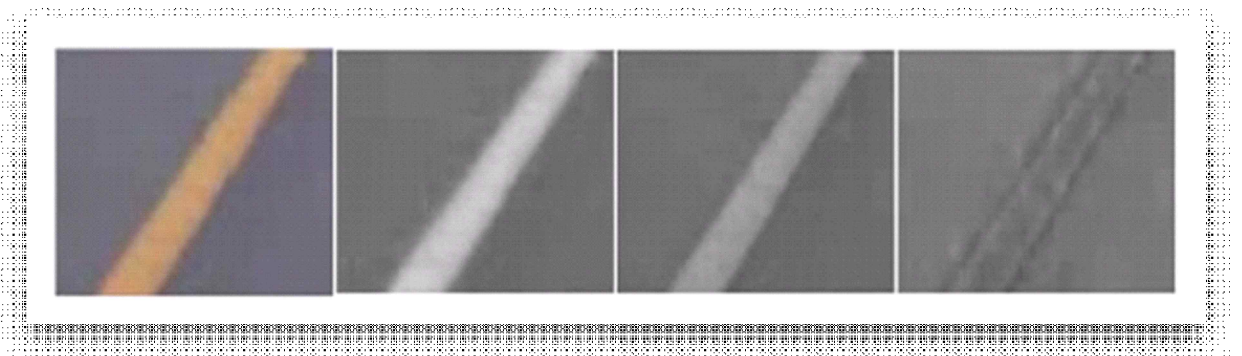


그림 8

2.3.2 표지판 인식

표지판은 안전하고 원활한 도로교통을 확보함과 동시에 도로이용자들의 안전하고 편안한 이

동을 제공하는 교통 소통 시설이다. 이런 표지판을 인식하여 원활한 교통 소통 및 무인 자동차가 올바른 이동을 할 수 있게 한다. 표지판을 인식하는 방법에는 OpenCV라이브러리를 이용하여 영상변환, 횡방향/종방향 연속셀 검출, 영상 합성, 영역확장, 특징점 추출, 템플릿 영상정합 등의 과정을 통하여 표지판을 인식하는 방법과 Retinex 알고리즘을 이용해 영상을 개선 시키고 HIS와 YCvCr의 두 가지 색상 체계에서 분할을 하여 표지판의 후보군을 추출하고, 히스토그램 평활화를 통한 명암비 개선을 한 뒤 검출된 표지판 영역에 대하여 일정 기울기를 회전하여 템플릿 매칭을 하는 방법등이 있다.



원본 영상



HSI 체계 이미지



YCbCr 체계 이미지



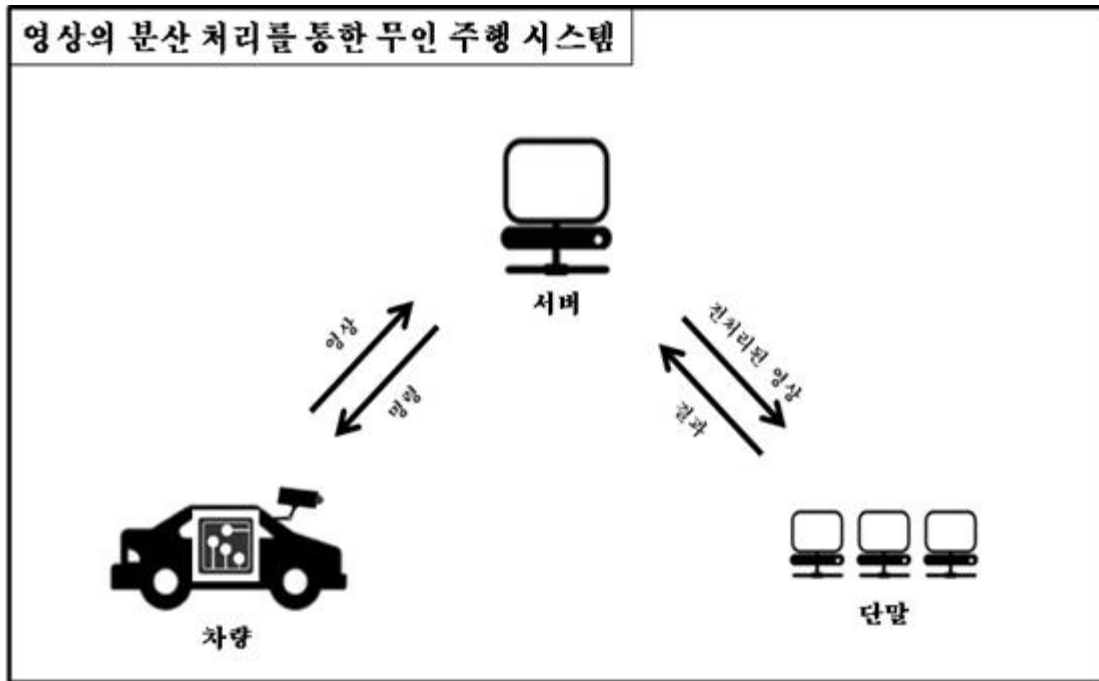
추출 된 이미지

2.4 분산처리

영상처리시 처리가 요구 되는 영상데이터의 크기에 비례하여 시스템의 부담은 더욱 더 증가하게 된다. 이런 시스템 부담을 덜어 주기 위해 현재 분산처리 시스템을 많이 이용하고 있다. 지금까지는 네트워크상에서, 클라이언트에서 만들어진 데이터를 서버에 실시간으로 전송하고, 이것을 서버에서 실시간으로 분석 및 처리하는 구조로 서버에 여러대의 클라이언트를 갖는 구조를 가졌지만 이는 서버에 처리할 데이터가 집중되어 데이터의 병목현상을 발생하게 된다. 병목현상이 발생하게 되면 시스템의 성능을 저하시키므로 현재 분산처리 시스템은 여러 대의 서버를 두어 처리하는 단순 분산처리 시스템을 구축하여 사용하고 있는 추세이다.

3. 영상의 분산 처리를 통한 무인 주행 시스템 개요

3.1. 전체 구조

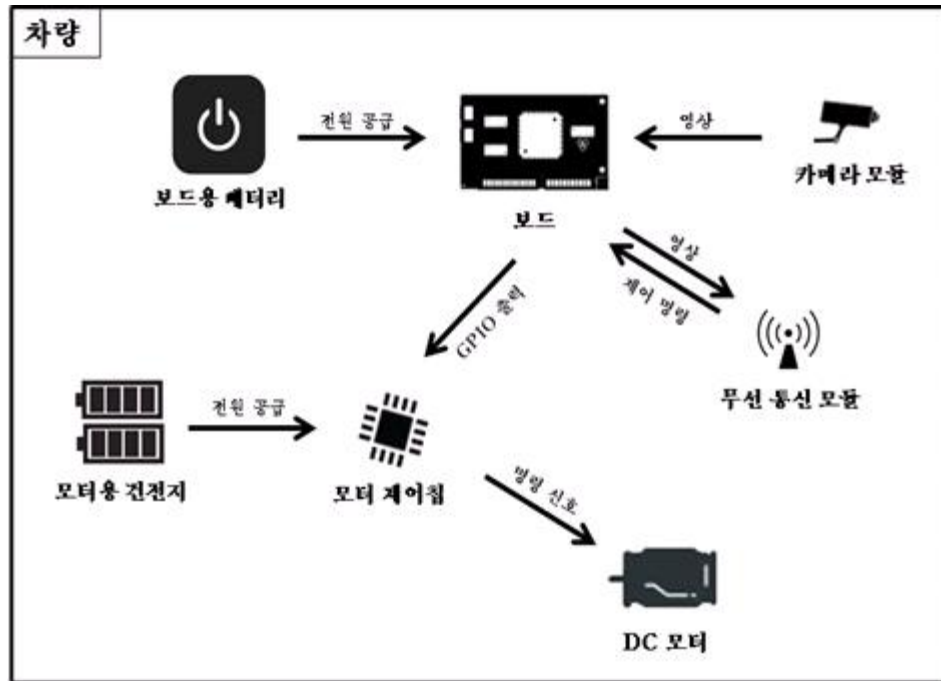


영상의 분산 처리를 통한 무인 주행 시스템은 차량, 서버, 단말로 구성된다. 차량에서는 영상을 촬영하고 이를 서버로 전송하고, 서버에서는 전송받은 영상을 단말들로 분배한다. 단말들에서는 분배받은 영상에 대해 지정된 작업을 수행하고 결과를 서버로 돌려준다. 서버에서는 결과들을 종합하여 차량에 적절한 명령을 내리게 된다.

초기 계획에서는 차량에서 영상 처리까지 모두 수행하려 했으나 한정된 하드웨어 자원으로 인해 원격지에서 영상 처리를 수행하는 것으로 노선을 변경하였다. 카메라로 영상을 촬영하여 전송하는 것만으로도 하드웨어 자원의 50% 이상이 사용되었다.

또한 영상 처리를 서버에서만 수행하지 않고 여러 단말들로 분배한 후 수행하는 분산 처리 방식을 적용하려고 한다. 이러한 분산 처리를 이용하면 하드웨어 자원을 많이 소모하는 영상 처리를 좀 더 빠르고 효율적으로 수행할 수 있다.

3.2. 차량 세부 구조



차량은 위 그림과 같은 요소들로 구성된다. 영상을 촬영하고 전송하기 위한 보드 외에도 차체를 움직이기 위해 모터를 제어하는 모터 제어칩이 존재한다. 모터 제어칩은 보드의 GPIO 출력 신호를 통해 모터를 작동시킨다. 보드와 모터 제어칩은 브레드 보드를 통해 연결되며 회로도 등 자세한 내용은 4장에서 설명한다.

보드와 모터는 요구 전압이 다르기 때문에 보드용 배터리와 모터용 건전지를 통해 따로 전원을 공급한다. 영상 촬영은 라즈베리 파이에서 지원하는 전용 카메라 모듈을 이용하며 무선 통신을 위해 USB 방식의 무선랜 카드를 추가로 장착한다.



차량 실제 모습



라즈베리 파이 B형



카메라 모듈

3.2.1. 보드

차량의 모든 동작을 제어하는 역할을 수행한다. 라즈베리 파이 모델 B를 이용한다.

3.2.1.1. 라즈베리 파이 모델 B

다양한 용도로 사용할 수 있는 단일 칩 시스템 컴퓨터. 700Mhz CPU와 512Mbytes RAM, GPU 또한 탑재하고 있으며 USB, GPIO, HDMI 등 다양한 입출력 형식을 지원한다.

3.2.1.2. 라즈비안

라즈베리 파이의 다양한 기능을 쉽게 사용할 수 있는 리눅스 기반 운영체제이다. 라즈베리 파이에 장착되는 SD 카드에 설치된다.

3.2.1.3. wiringPi

라즈베리 파이에서 GPIO를 쉽게 사용할 수 있도록 해주는 오픈 소스 라이브러리이다. 모터 제어칩으로 신호를 보내기 위해서 GPIO를 사용해야 하므로 설치한다. github에서 받을 수 있다.

3.2.2. 카메라 모듈

라즈베리 파이에서 영상을 촬영하기 위해 이용하는 장치. USB 웹캠도 사용할 수 있지만 라즈베리 파이용 카메라 모듈이 사용하기 더 간편하다. 1080p 동영상을 초당 30 프레임으로 촬영 가능하며 정지영상은 2592x1944 해상도까지 지원한다. 무게는 케이블 포함 3.4g으로 매우 가볍다.

3.2.2.1. openCv

실시간 영상 처리에 중점을 둔 오픈 소스 라이브러리이다. 객체 인식, 안면 인식, 템플릿 매칭 등 다양한 기능을 제공한다.

3.2.2.2. raspicam_cv

촬영한 영상을 openCv 라이브러리에서 사용할 수 있도록 변환해주는 오픈 소스 라이브러리. 차량에서 촬영한 영상을 openCv의 IplImage 객체로 변환하여 서버로 전송한다.



무선 랜 카드



보드용 배터리

3.2.3. 무선 통신 모듈

라즈베리 파이에 무선랜이 내장되어 있지 않기 때문에 USB 무선 랜카드를 이용한다.

3.2.4. 보드용 배터리

라즈베리 파이에 전원을 공급하는 장치로 스마트폰 보조 배터리를 이용한다.



모터 제어칩



DC 모터

3.2.5. 모터 제어칩

DC 모터는 라즈베리 파이에서 GPIO를 통해 직접 제어할 수 없다. L293D라는 칩을 이용하면 모터의 전원 on/off 와 회전 방향을 라즈베리 파이에서 GPIO를 통해 제어할 수 있다. 칩 하나에서 최대 두 개의 모터를 제어할 수 있다.

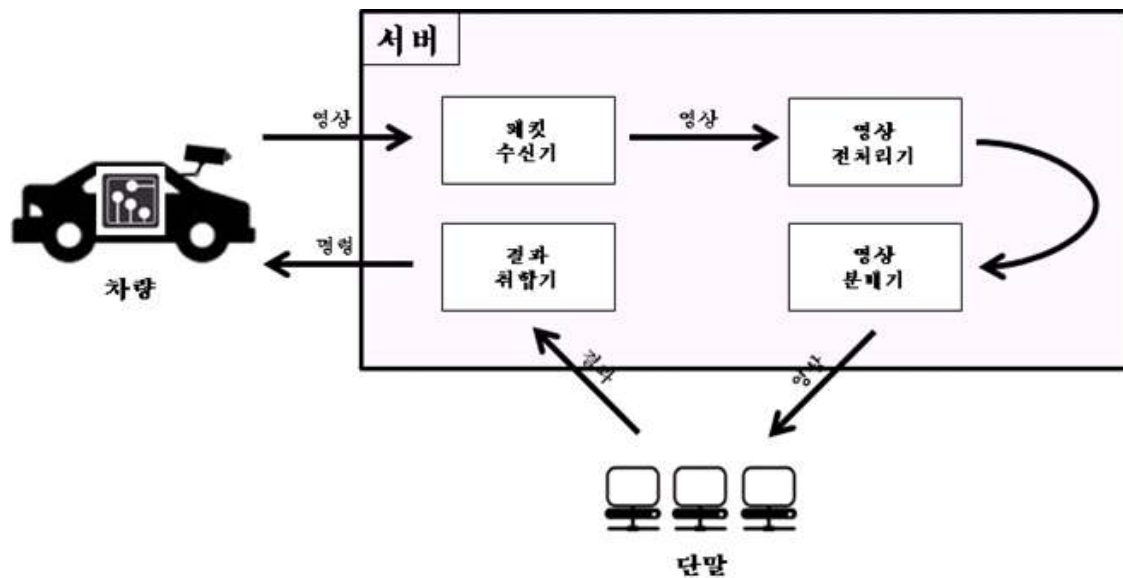
3.2.6. 모터용 건전지

DC 모터가 동작하기 위한 전력을 공급한다. 1.5V 일반 건전지 4개를 직렬로 연결하여 6V의 전력을 공급한다.

3.2.7. DC 모터

전력이 연결되면 전류의 방향에 따라 정해진 방향으로 회전하는 일반적인 모터이다.

3.3 서버 세부 구조



3.3.1. 전체 구조

서버는 차량으로부터 전송 받은 영상을 단말로 분배 후 영상을 처리해 수행해야 할 명령을 종합하여 다시 차량으로 전송하는 역할을 위해 4부분으로 구성 되어 있다.

3.3.2. 패킷 수신기

차량으로부터 지속적으로 들어오는 동영상의 패킷을 수신하는 부분으로 하나의 스레드로 구성된다. 패킷 수신기에서 수신한 영상의 패킷을 이미지별로 영상 전처리기쪽으로 전달 한다.

3.3.3. 영상 전처리기

영상을 각 단말로 보내 탐색하기 전에 빠른 탐색을 위한 전처리 알고리즘을 적용하여 영상을 처리하는 부분이다. 영상 전처리기에서 처리한 영상을 각 단말로 전송하게 된다.

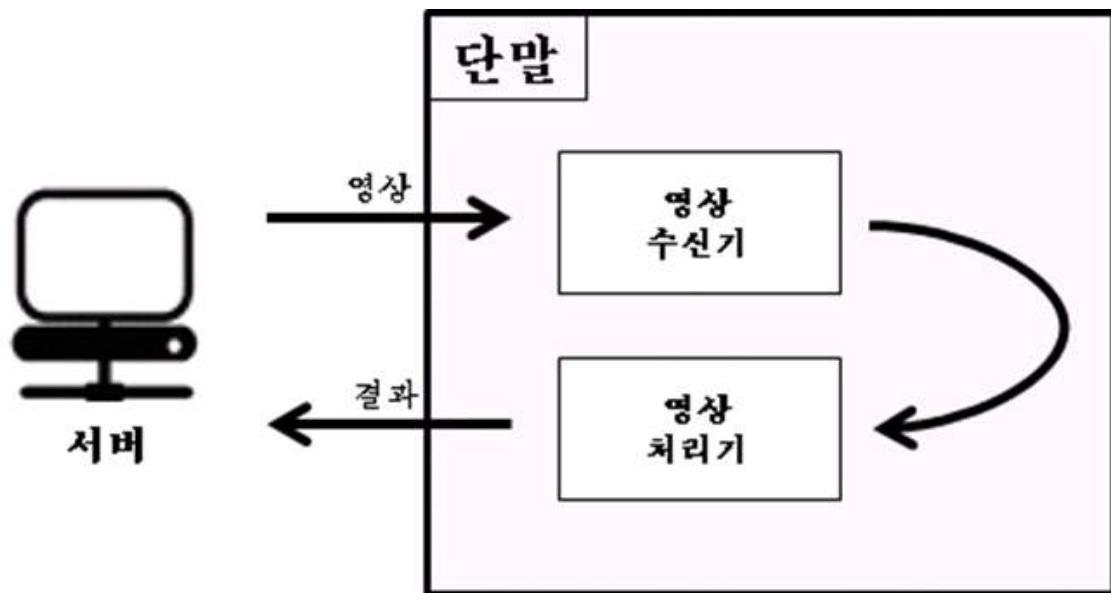
3.3.4. 영상 분배기

영상 전처리기에서 처리한 영상을 각 단말로 각각 전송하는 부분으로 각 단말별로 영상을 전송한다. 영상의 순서를 맞추고, 전송의 손실을 방지하기 위해 tcp 소켓으로 구성한다.

3.3.5. 결과 취합기

각 단말에서 처리한 결과를 종합해 차량이 수행해야 할 명령을 결정한다. 각 단말별로 결과를 전송받기에 각각 스레드로 구성되며 종합한 결과를 차량으로 전송한다.

3.4 단말 세부 구조



3.4.1. 영상 수신기

서버로부터 들어오는 영상의 패킷을 수신하는 부분으로 스레드로 구성된다. 전송받은 영상은 처리를 위해 영상 처리기로 전달한다.

3.4.2. 영상 처리기

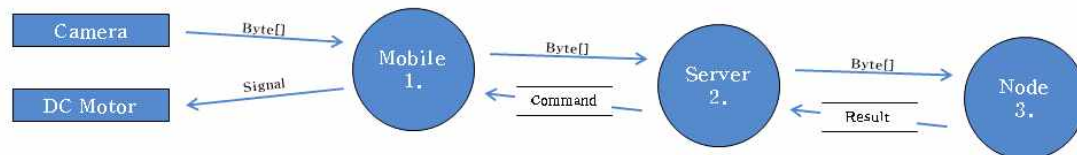
서버로부터 전달받은 영상을 여러 처리 알고리즘을 이용하여 처리 후 다시 서버로 결과를 전송하는 부분이다. 영상의 처리는 영상을 통해 차량이 수행해야 할 명령을 구하기 위한 처리로 몇 가지 템플릿 매칭 알고리즘을 이용한다. 아래 사진은 서버로 가상머신 환경에서 linux를 운영체제로 이용한다.



서버

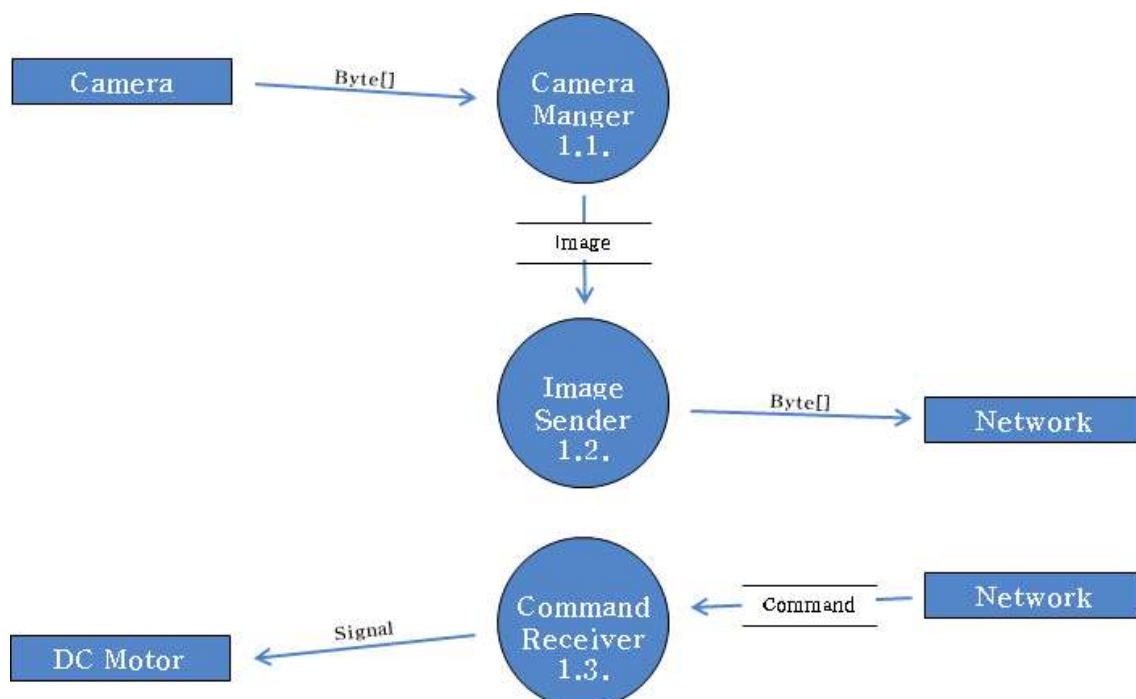
3.5. DFD

3.5.1. Total System's DFD Level 1

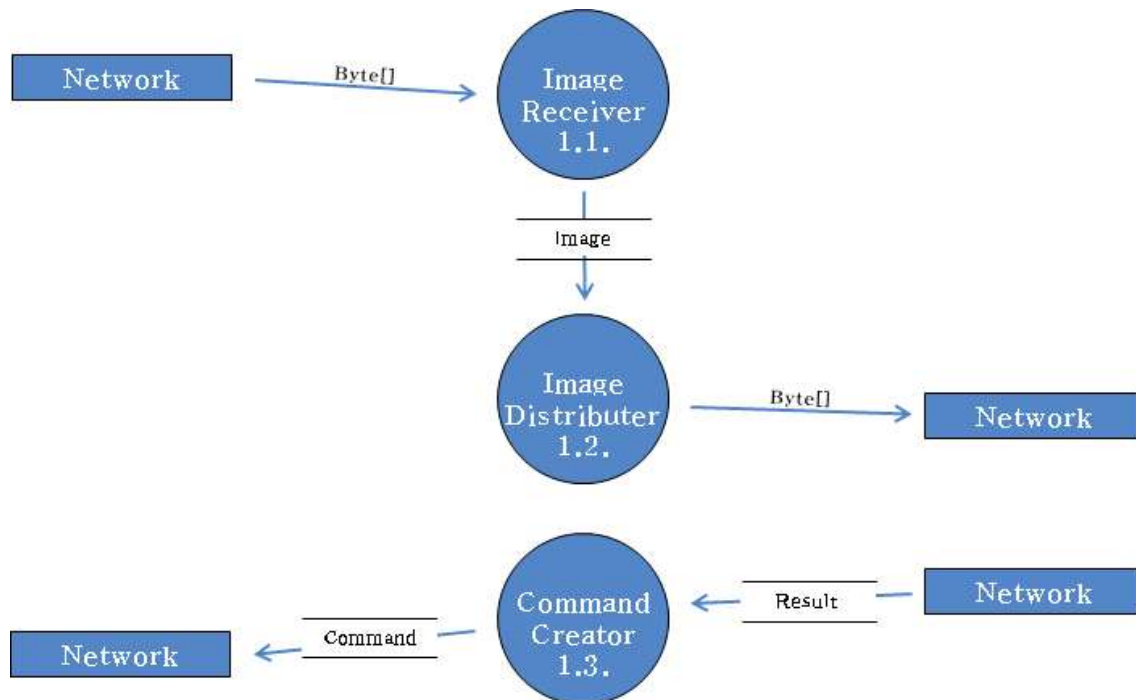


3.5.2. DFD Level 2

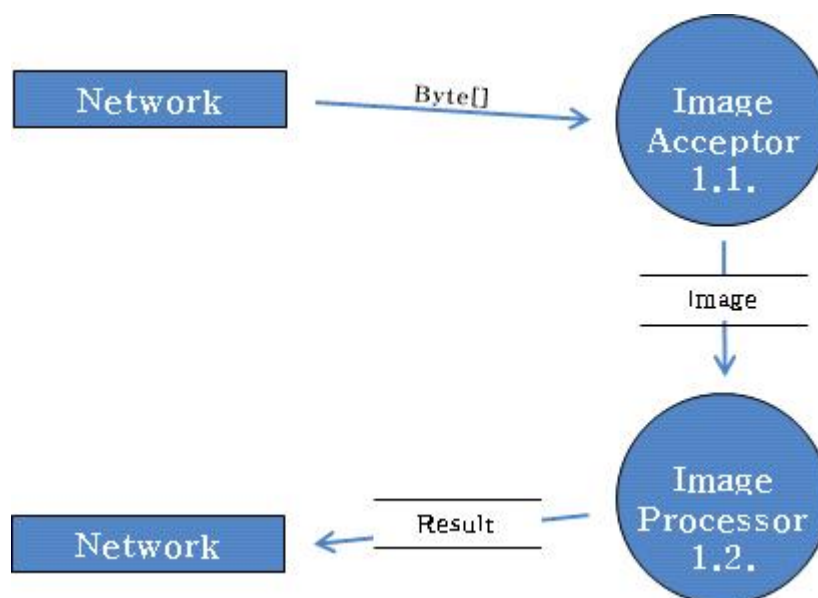
3.5.2.1. Mobile



3.5.2.2. Server



3.5.2.3. Node



4. 차량 하드웨어 구현

현재까지 본 시스템의 구성 요소 중 가장 많이 진행된 차량 부분의 구현 방식을 설명하고자 한다. 차량에서 가장 중요한 부분은 시스템이 모터를 제어할 수 있도록 하는 것이다. 일반적으로 DC 모터는 전원을 공급하면 동작하고 차단하면 동작을 멈춘다. 그러나 라즈베리 파이에서 GPIO 출력으로 줄 수 있는 전기 신호는 최대 5V이기 때문에 모터가 동작할 만큼 충분한 전원을 공급할 수 없다. 따라서 L293D 칩을 경유하여 모터를 제어한다.

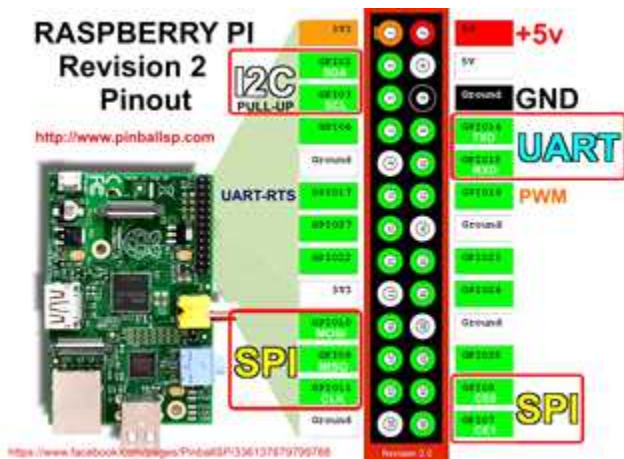


그림 29

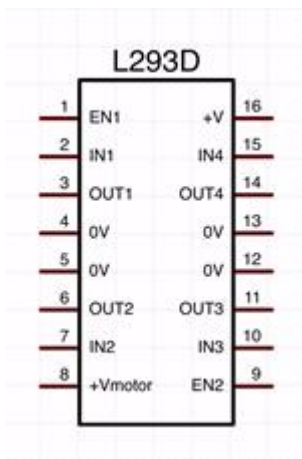


그림 30

먼저 라즈베리 파이에서 GPIO로 L293D에 입력 신호를 보낸다. L293D는 입력 신호에 따라 설정된 방향으로 건전지에서 공급받은 전력을 모터와 연결시켜 준다. 이를 위해 회로는 아래와 같이 구성한다.

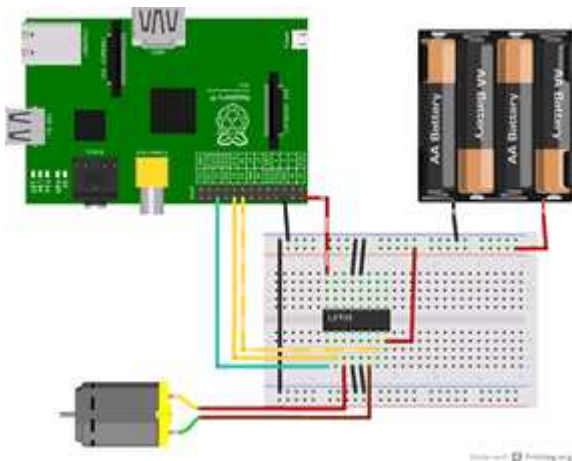


그림 31

5. 특징 및 활용 방안

5.1 특징

5.1.1 하드웨어

자동차 제어를 위해 메인 보드로는 라즈베리파이를 사용한다. 라즈베리파이를 이용해 자동차 구동과, 카메라 제어, 통신 및 명령어 처리와 같은 전반적인 사항을 제어 할 수 있도록 운영 체제를 설치하여 활용한다. 자동차 구동을 위한 구동부는 dc모터와 모터 컨트롤러를 이용한다. 모터 컨트롤러는 dc모터를 쉽게 컨트롤 하기 위한칩셋으로 모터 컨트롤러의 컨트롤은 라즈베리 파이에서 직접 담당한다.

5.1.2 영상처리

자동차의 무인 이동 방식의 동작에는 추가적인 센서 사용 없이카메라를 통해 들어오는 영상만을 이용해 처리한다. 카메라의 영상을 이용하여 자동차의 방향 및 동작을 결정할 수 있도록 영상을 처리한다. 영상 처리의 결과를 이용해 전진, 후진, 회전, 장애물 회피등의 동작을 수행 한다. 영상 처리에는 템플릿 매칭 등 여러 알고리즘을 사용하여 거리 파악 및 동작을 파악 한다.

5.1.3 분산처리

하나의 하드웨어(머신)에서 모든 영상을 처리하게 될 경우 기대만큼의 성능을 보이기 어렵다. 그래서 하드웨어의 처리속도 한계를 극복하기 위하여 하나의 머신에서 직접 영상을 모두 처리하기 보다는 영상의 처리를 여러 노드로 분산시켜 처리한다. 카메라를 통해 들어오는 영상을 서버로 전송하고 서버에서는 영상을 각 노드로 분산해 처리 후 결과를 종합하여 전송하는 방식이다. 대상이 되는서버는 하나이고 서버와 연결되어 처리 하는 노드의 경우 동적으로 개수를 변경 가능 하도록 구현한다.

5.2 활용 방안

5.2.1 하드웨어

소형 무인자동차를 제작 하여 작은 실험환경에서 무인 자동차를 쉽게 테스트 할 수 있다. 실제 테스트를 위하여 많은 노력과 비용이 들어가는 무인 자동차 시스템을 우선 실험 환경을 통해 실험 함으로써 다소 적은비용과 노력으로 문제들을 해결해 나갈 수 있다. 또한 같은 방식의 보드를 통해동작을 제어해야 하는 무인 항공기, 탐사 로봇 같은 다른 무인 시스템과 그 이외의 IOT 등과 같은 분야에 하드웨어 및 통신부분을 적용하여 제작하고 활용 및 테스트를 위해 사용할 수 있다.

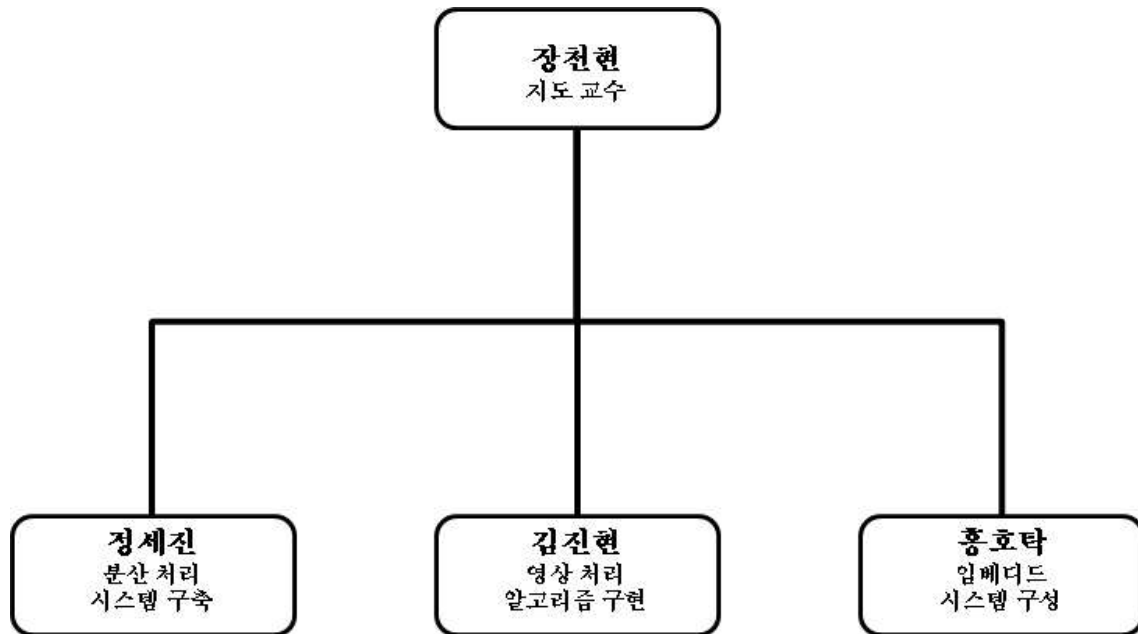
5.2.2 영상처리

영상만을 가지고 처리 함으로써 부가적인 센서 하드웨어 사용이나 센서 신호 처리 등의 작업이 필요없이 카메라만을 이용하여 제작하기 때문에 비용 절약의 효과를 기대 할 수 있으며 더 쉽게 제작 할 수 있는 부분이 존재한다. 또한 무인 자동차뿐만 아니라 다른 종류의 무인기체 또는 객체 인식이 필요한 다른 분야에 현재 사용한 객체인식 알고리즘 및 내용을 활용할 수 있다.

5.2.3 분산처리

많은 cpu 자원을 요구하는 영상처리에 대해서 여러 노드를 통해 분산시켜 처리함으로써 처리 속도 향상을 기대 할 수 있다. 분산 처리 시스템을 구축할 경우 처리할 대상 정보의 네트워크 전송이라는 오버헤드가 존재 하지만 낮은 클럭의 단일 cpu에서 처리 하는 것 보다 높은 클럭의 머신에서 처리하는 것이 그리고 여러 머신(cpu)에서 처리하는 것이 속도 향상을 기대할 수 있다. 또한 앞으로 처리속도가 중요한 시스템 개발에 현재 결과를 응용하여 여러 분야에서 사용할 수 있다.

6. 역할 분담



6.1. 정세진

영상의 전송 및 분배, 처리를 위한 분산 처리 시스템을 구축한다.

6.2. 김진현

영상을 처리하여 객체를 인식하고 상황을 판단하는 알고리즘을 구현한다.

6.3. 홍호탁

영상의 촬영 및 차량이 명령에 따라 올바른 작업을 수행하도록 한다.

7. 일정 계획

번 호	내 용	일 정							기간 (주)
		5	6	7	8	9	10	11	
1	자료 조사 및 분석	■							2
2	제안서 작성	■							2
3	시스템 설계		■						2
4	프레임 제작			■	■				4
5	임베디드 환경 구성			■	■				4
6	영상 처리 알고리즘 구현				■	■	■		8
7	분산 처리 환경 구축					■	■		6
8	모듈 통합 및 시스템 검증						■	■	4
9	최종 보고서 작성							■	3

8. 참고자료

손영선, 신일식, 박상열, "픽셀 수치 정보를 이용한 교통안전 표지판 인식", KFIS Spring Conference, 2007

나원섭, 엄성제, 정용진, "템플릿 매칭을 이용한 교통 표지판 인식 시스템", 동계종합학술 발표회, 2014

남기환, 배철수, "교통표지판 인식을 위한 비전시스템, 한국해양정보통신학회논문지.2004

김기홍, 정규수, 윤준희, "OpenCV를 이용한 도로표지 영상에서의 방향정보 자동인식", Journal of the Korean Society of Surveying, 2013

전향수, "무인자동차 기술 및 개발 동향", 정보통신산업진흥원, 2012

공영일, "구글카 사업 동향과 전개 방향", 정보통신정책연구원, 2013

장규태, 정차근, "주행 차선검출을 위한 영상처리 고속화 방법", 대한전자공학회, 2007

송정영, "차선 검출에 의한 자동 주행로 판단 알고리즘", 한국정보과학회, 2007

S. M. Akramullah, I. Ahmad, M. L. Liou, "Performance of Software-Based MPEG-2 Video Encoder on Parallel and Distributed Systems", IEEE, 2000

영상 보행자 검출 기술

<http://darkpgmr.tistory.com/53>

라즈베리파이 서보모터 제어

<http://www.rasplay.org/?p=6783>

wiringPI를 이용한 GPIO

<http://embejide.tistory.com/39>

RaspiCam: 라즈베리파이 카메라

<http://www.uco.es/investiga/grupos/ava/node/40>

라즈베리파이 GPIO

<http://www.rasplay.org/?p=2049>

<http://www.rasplay.org/?p=3241>

리눅스 OpenCV 설치

<http://blog.danggun.net/2121>