

단일 카메라를 사용한 독립형 자율이동로봇 개발

A study on stand-alone autonomous mobile robot using mono camera

정 성 보, 이 경 복, 장 동 식

Sung-Bo Jung, Kyung-Bok Lee and Dong-Sik Jang

고려대학교 산업시스템정보공학과

Department of Industrial Systems and Information Engineering
Korea University

요 약

본 논문은 실제 무인주행자동차에 적용할 수 있는 비전 기반의 소형 자율이동로봇 개발에 관한 연구를 제시한다. 이전의 자율주행차량은 하드웨어 설계의 복잡성, 실장의 어려움과 많은 계산량으로 인해 PC에 대한 의존도가 높았다. 본 논문에서는 고속에서 정확한 조향 및 빠른 이동을 할 수 있고, 단일 카메라를 사용한 독립형 시스템으로 지능적 인식을 할 수 있는 소형 자율이동로봇을 제안한다. 제안된 시스템은 폭 25~30cm, 총길이 200cm로 만들어진 트랙에서 실험하였다. 실험 로봇은 직선 트랙에서 평균 32.9km/h, 곡률반경 30~40m인 곡선트랙에서 평균 22.3km/h의 속도로 주행할 수 있었다. 이 시스템은 실제 무인 자동차를 쉽게 만들기 위해 사용할 수 있는 차선 인식 알고리즘을 적용한 소형 자율이동로봇 시스템에 대한 하나의 모델을 제시할 수 있었다.

Abstract

This paper introduces a vision based autonomous mini mobile robot that is an approach to produce real autonomous vehicle. Previous autonomous vehicles are dependent on PC, because of complexity of designing hardware, difficulty of installation and abundant calculations. In this paper, we present an autonomous mobile robot system that has abilities of accurate steering, quick movement in high speed and intelligent recognition as a stand-alone system using a mono camera. The proposed system has been implemented on mini track of which width is 25~30cm, and length is about 200cm. Test robot can run at average 32.9km/h speed on straight lane and average 22.3km/h speed on curved lane with 30~40m radius. This system provides a model of autonomous mobile robot adapted a lane recognition algorithm in order to make real autonomous vehicle easily.

Key Words : 자율 이동 로봇, 주행형 이동 로봇, 영상 인식, 지능 로봇, 차선 인식, Stand alone

I. 서 론

최근 빠른 속도로 발전된 로봇 산업은 다양한 분야에서 응용 또는 실생활에서 쓰이고 있다. 로봇 공학의 한 부류로써 자율이동로봇은 인간의 개입 없이 스스로 주어진 목적지까지 주행 할 수 있는 로봇을 말한다. 이동로봇은 이동 형태에 따라 두 발을 사용하여 걸어 다니는 보행형 이동로봇(Humanoid)과 바퀴를 사용하여 이동하는 주행형 이동로봇으로 나눌 수 있다. 본 논문은 네 바퀴를 사용하는 소형 자율이동로봇에 관한 연구로서 실제 자율주행자동차(Autonomous Vehicle)에 근접한 소형 자율이동로봇 시스템을 개발하는데 있다. 본 시스템은 단일 카메라를 사용한 영상 기반의 독립형(Stand alone) 자율이동로봇을 개발하는 것인데, 본 연구에서 개발한 자율이동로봇은 크게 영상

인식시스템과 구동제어시스템, 센서제어시스템의 세 부분으로 구성된다.

본 연구에서는 자율이동로봇의 주행을 위한 차선의 인식 방법과 올바른 조향 각도를 계산하는 방법에 대해 제안한다. 자율이동로봇이 주어진 차선을 자율주행하기 위해서는 먼저 로봇의 전방을 볼 수 있도록 이미지 센서인 CCD 카메라를 로봇에 장착하고 이를 통해 입력된 주행 영상을 이용하여 차선을 인식하도록 하였다. 그 다음 인식된 차선을 이용하여 자율이동로봇의 방향을 계산하는 방법에 의해 상황에 맞는 조향의 회전 각도를 산출한다. 또한 로봇 전방에 장착된 초음파 센서를 통해 전방의 장애물이 있을 경우 정지할 수 있도록 구성하였다.

자율이동로봇에 장착된 CCD 카메라를 이용하여 차선을

인식하는 방법은 기존 연구를 토대로 크게 두 가지로 나누어 볼 수 있다. 대표적인 방법은 입력된 영상을 전부 처리하여 각각의 픽셀마다 입력받은 영상이 특징 부분에 속하는지를 판단하는 것이다. 이 방법은 영상을 처리하기 위해 많은 연산량이 필요하기 때문에 PC에 대한 의존도가 높은 편이다. 예를 들어 CCD 카메라로부터 받은 영상을 회색음영(Grayscale) 360*240의 크기로 처리할 경우 하나의 프레임을 처리하는 메모리 크기는 86,400 Byte가 필요하다. 실시간으로 영상 처리를 하기 위해선 보통 하나의 프레임을 처리하는 게 아니라 초당 30프레임 정도의 영상을 처리해야 한다. 이렇게 되면 2,592,000 Byte의 메모리가 필요하게 된다. 독립형(Stand alone)기반으로 구현하면 연산 속도는 빠르나 사용할 수 있는 메모리 크기가 PC 기반에 비해 현저히 적은 편이다. 만약 독립형(Stand alone)기반에 전처리(Preprocessing)를 한다면 적은 메모리로 인해 시스템에 과부하가 생겨 기기가 다운되거나 처리 속도가 매우 느려지기 때문에 실 연구에 도움을 주지 못한다. 이러한 문제점을 보완하기 위해서는 입력받은 영상을 회색음영(Grayscale)으로 바꾸어 차선을 인식하는 방법이 있다. 본문에서 제시하는 방법은 독립형(Stand alone)기반의 실시간 처리에 초점을 둔 방법으로 입력된 영상의 특징점을 처리하여 차선을 인식하는 방법이다. 그리고 입력된 영상으로부터 추출된 특징점인 차선으로부터 로봇의 조향각을 결정하기 위한 방법을 사용하였다. 그 방법으로 로봇의 회전 궤적을 이용하여 가고자 하는 실 좌표계상에서의 위치를 구한 다음 카메라의 기하학적인 관계를 고려함으로써 핸들의 조향각을 결정하게 되는 카메라의 기하학적인 모델을 이용하는 방법을 사용하였다.

II. 시스템 구성

본 연구를 통해서 개발한 자율이동로봇은 영상을 기반으로 전방의 차선을 인식하여 정해진 차선을 따라 자율적으로 주행하는 비전을 이용한 소형 주행로봇이다. 그러나 단순히 차선만을 따라 주행하는 것이 아니라, 적외선 센서를 이용하여 전방의 장애물을 인식하여 장애물을 피해가거나 정지할 수 있으며, 스스로 차선을 보고 진행 방향을 판단하여 자율적으로 주행 및 이동하는 자율이동로봇이다.

본 자율이동로봇은 영상인식부, 구동제어부, 센서제어부, 통신제어부로 구성된다. 영상인식부는 카메라로부터 얻은 전방 영상에서 차선을 인식하여 로봇이 진행해야 할 실시간 방향을 판단하여 차선을 따라 주행할 수 있도록 구동제어부에 지시를 내리는 역할을 한다. 구동제어부는 영상인식부로부터 받은 명령을 바탕으로 조향과 속도를 제어하는 일을 담당한다. 센서제어부는 로봇 전방에 장착한 초음파 센서를 통하여 전방의 장애물을 감지하여 제어하는 일을 담당한다. 그리고 통신제어부는 각 제어부 간의 통신 및 서버와의 무선 통신을 담당한다.

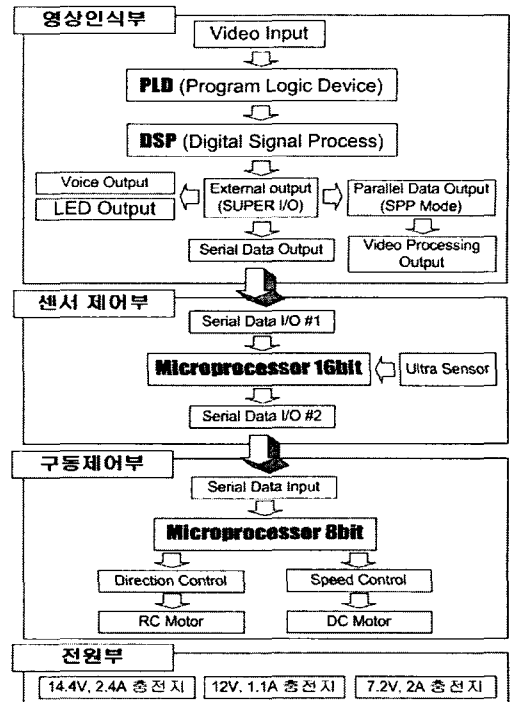


그림 1. 시스템 전체 구조
(Fig. 1) A structure of AMR system

1. 영상인식부

영상인식부는 흑백 전용 CCD 카메라와 Texas Instrument (이하 TI)사의 DSP TMS-320VC33을 이용한 영상처리보드로 구성된다. CCD 카메라는 가로29mm×세로29mm×높이28mm의 초점거리 3.6mm인 소형 카메라이다. 영상처리보드는 NTSC 영상을 디코딩할 수 있는 Philips사의 SAA7111H Decoder Chip을 사용하였다.

영상 처리보드의 메인 CPU는 TI사의 DSP TMS320C33을 사용했으며, 영상 디코더는 Phillips사의 SA7111AH를 사용하였다. 그리고 디코더에서 넘어온 동기 신호를 이용하여 각각 Pixel의 영상 데이터를 32bit로 정렬해서 DSP가 가져갈 수 있도록 Lattice사의 PLD(Program Logic Device)를 사용하였다. 그리고 영상처리를 위한 메모리를 1M(128K×8bit)bit RAM을 사용하였는데 어드레스 신호 32bit를 맞추기 위해 4Mbit를 사용하였다.



그림 2. 영상인식부 (카메라, 장착 모습, 영상처리보드)
(Fig. 2) Vision recognition system

영상처리에 CCD 카메라로부터 NTSC 신호가 입력이 되면 SA7111AH에서 각 Pixel값을 수평신호와 수직신호에 맞춰서 출력해 준다. 본 자율주행로봇은 회색음영(Grayscale)영상을 사용하므로 YUV 형식의 신호 중

Y(Luminace) 값을 처리한다. 그리고 SA7111AH에서 나온 8bit 영상신호를 EPLD에서 32bit씩 묶어 DMA 전송으로 SD RAM으로 데이터가 옮겨진다. 한 화면의 크기는 360 × 240이며, 전체 크기는 172,800 Pixel이다.



그림 3. 영상처리보드 구조
(Fig. 3) A structure of image processing board

2. 센서제어부

센서제어부는 여러 개의 센서로부터 입력을 받기 위해 센서 입력을 독립적으로 받고 처리하며 계산할 수 있는 보드를 그림 3과 같이 설계했다. 16bit 범용 컨트롤러로 본 논문에서는 초음파 센서 한 쌍을 이용하였다. 센서제어부의 구동 흐름은 DSP 보드에서 차선을 인식한 처리 결과 값을 시리얼 통신으로 센서 제어부로 전송한다. 센서 제어부는 전방의 장애물을 초당 10회 정도 찾는다. 만약 장애물을 검지할 경우에 센서 처리부에서 자체적으로 만든 패킷 데이터를 구동제어부로 보내어 로봇의 진행 방향을 바꾸거나 로봇을 정지하게 한다.

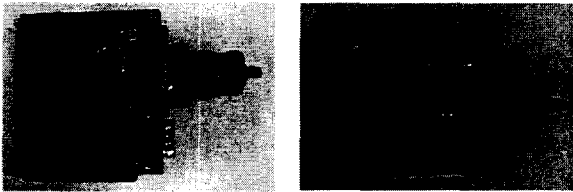


그림 4. 센서제어부 와 구동제어부
(Fig. 4) Sensor control system & mobile control system

3. 구동제어부

구동제어부는 방향을 제어하는 서보 모터와 속도를 제어하는 DC모터를 제어하기 위해 Atmel사의 8bit 마이크로 프로세서인 AVR163L을 이용하였다. 서보 모터를 제어하기 위해서는 PWM 주기로 제어를 한다. 80C196은 DSP에서 명령을 받은 다음 센서로부터 값을 확인한 후 AVR163L에게 명령을 보낸다. 이를 통해 전방의 장애물을 검지하고 조향 및 속도를 제어하게 되는데 전체 시스템이 동작하는데 시스템지연이 없도록 설계하였다.

가. RC 모터 제어

로봇의 방향을 전환하기 위한 액추에이터로 DC모터, RC모터를 들 수 있다. 실제 무인자동차에서는 DC모터와 엔코더를 이용해 정확한 DC 모터 회전수를 제어 할 수 있다. 실제 무인자동차에서는 모터의 크기와 무게가 그리 문제가 되지 않겠지만 소형 자율이동로봇은 소비 전력 및 무게, 크기에 민감하다. 시중에서 구할 수 있는 소형 DC모터와 소형엔코더를 DC모터 뒷축에 붙이면 되지만, 이 역시 크기가 커서 소형 자율 이동로봇에 장착하기가 어려웠다. 그래서 무선 조종 완구 등에 많이 사용되는 소형 서보 모터를 로봇에 장착하였다. 소형 서보 모터란 Radio Controlled Servo Motor의 약자로 보통 무선 조종 완구 등

에 많이 사용되는 서보 모터이다.

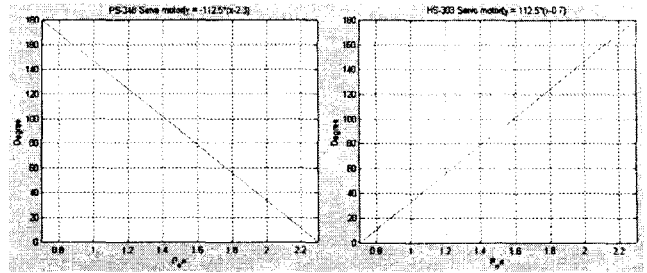


그림 5. RC모터의 분해능과 속도 그래프
(Fig. 5) Graph of resolution of RC motor & Speed

RC 모터의 내부 구조는 기어와 위치 검출용 가변저항(Potentiometer)이 함께 포함되어 있으며 외부의 PWM신호로 일정 주기의 PWM 신호를 주면 모터가 왼쪽 0도에서 오른쪽 180도 사이를 움직인다. RC 모터 신호선은 각각 Vcc, Gnd, PWM선이다. 먼저 자율이동로봇에 모터를 장착하기에 앞서, 모터의 분해능과 속도등을 테스트 해 보았다. 속도를 제어하는 DC모터는 정지상태인 0부터 최고 속도인 20까지 0~20사이의 계수로 분해하였고, 방향을 제어하는 RC모터는 로봇 방향을 중심으로 삼았을때 중심을 기준으로 -120 ~ 0 ~ 120으로 세밀히 분해하였다.

4. 통신제어부

통신제어부는 각각의 제어부 간의 통신을 한다. 통신 순서는 영상 처리부, 제어부, 방향·속도제어부 순으로 구성된다.

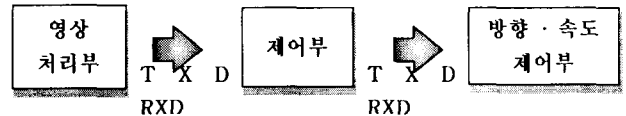


그림 6. 통신제어부 구조
(Fig. 6) A structure of communication controller

통신 프로토콜은 Physical Layer, Transport Layer, Control Layer로 3개의 Layer층으로 구성되며, Data Packet의 구성은 2개 바이트(Byte)의 크기이다.

7	6	5	4	3	2	1	0
Flag	Right/Left	Speed	Stop	DATA7	DATA6	DATA5	DATA4
7	6	5	4	3	2	1	0
Flag	Right/Left	Speed	Stop	DATA3	DATA2	DATA1	DATA0

표 1. 통신 프로토콜 패킷 포맷
(Table. 1) Packet Format

5. 전원부

전원부는 전체 시스템에 안정된 전원을 공급하기 위한 부분으로 충전지를 고려해서 12V에 맞춰서 구성하였다. 보드에 아날로그 부분이 존재하므로 Digital 잡음이 Analog 부에 영향을 주지 않도록 배터리를 분리하여 구성하였다. 충전지는 세개를 이용하였는데 방향·속도 제어부 전원을

12V · 1.1A, DC 모터 제어부에는 7.2V · 2A, 소비전력이 많은 영상처리부와 카메라는 14.4V · 2.4A 를 사용하였다.

방향 · 속도제어부는 RC 모터를 제어해야 하므로 잡음이 심해서 영상처리부와 제어부의 전원을 각각 독립시켰다.

III. 주행 제어 알고리즘

자율이동로봇은 크게 영상인식, 위치인식, 구동제어, 통신 제어 등 크게 네 가지 알고리즘으로 구성된다.

영상인식은 카메라로부터 얻은 영상에서 차선을 추출한 후, 차선의 방향을 인식하여 주행방향을 결정한다. 결정된 방향이 구동제어부로 전달되면, 구동제어부는 센서로부터 센서 값에 해당하는 패킷의 정보를 받게 된다. 이렇게 영상 처리로 인식한 장애물의 신호와 센서로부터 나오는 신호를 때때로 독립시켜 장애물에 대한 추돌을 미연에 예방 할 수 있도록 한다. 그리고 바퀴의 속도와 방향을 제어하여 진행 방향을 조절한다. 본 실험에서는 부정확한 차선이더라도 영상인식에서 차선을 인식하여 진행할 수 있는 방향을 인식하고, 위치인식에서 결정한 목적지까지의 방향을 바탕으로 최종 주행 방향을 결정한다. 통신제어는 각 제어부간에 통신하는 프로토콜을 구현하였다.

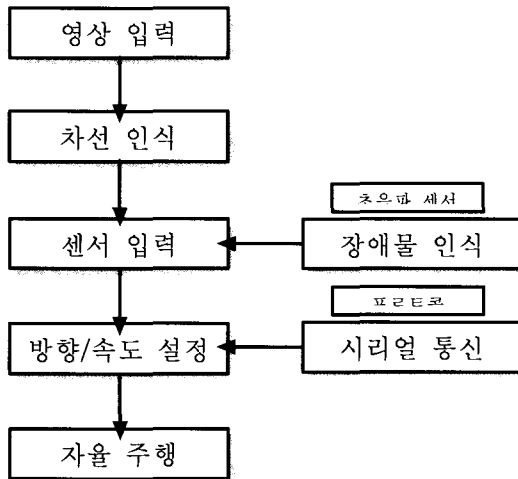


그림 7. 자율이동로봇의 전체 알고리즘
(Fig. 7) Algorithm of AMR

1. 차선 인식

가. 영상입력

NTSC 방식 카메라 전체 해상도는 720×480인데, 본 연구에서는 영상처리를 위해서 전체 해상도의 1/4 즉, 360×240의 영상을 사용한다. 영상이 크다면 차선 분해능이 높아지는 등의 이점이 있다. 하지만 720×480의 Full frame 영상을 처리하려면 비디오 신호 입력을 받아서 한 화면씩 미리 저장해 놓는 장치가 없는 한 13ms 즉, Odd field와 Even field가 모두 저장된 후 다음 영상이 나올 때까지 Blank 신호가 나오는 시간 동안 모든 알고리즘이 실행되어야 한다. 뿐만 아니라 화면 크기가 크기 때문에 1/4의 영상을 처리 했을 때보다 최소 4배의 메모리가 필요하게 된다.

따라서 본 연구에서는 계산량을 줄이기 위해서 수평 해상도를 360(표준해상도 720의 절반)으로 낮추고, Odd Field만을 처리하는 방법을 사용했다. 이렇게 화면 크기를 줄여서 사용하면 한 field가 출력되는 시간과 2번의 Blank 신호가 나오는 시간 즉, 17.9ms (=15.3ms+2×1.3ms) 동안 영상 처리에 필요한 알고리즘을 수행할 수 있다. CPU 성능에 따라 다소 차이는 있겠지만, CPU Instruction Cycle을 33ns 라고 하면 542,424 (=17.9ms/33ns)번 즉, 약 54만번의 연산이 가능한 시간이다. 360×240 크기의 화면은 86,400개의 pixel로 이루어져 있고, 따라서 6번 정도의 리드타임(read time)을 확보할 수 있다. 따라서 실시간 영상처리가 가능하다.

나. 차선 추출

본 연구에서는 차선을 추출하기 위해 자율이동로봇의 전방을 볼 수 있도록 CCD 카메라를 설치하여 주행로의 영상을 획득하도록 하였다. 실제 주행로상의 차선은 직선과 다양한 곡률의 곡선으로 이루어진다. 하지만 로봇 전방에 나타나는 차선의 형태는 거의 직선의 형태를 띠며 또한 차선의 폭이 크고 차선의 색깔도 명확하다. 따라서 이 부분의 차선을 찾는 것은 원거리의 차선을 찾는 것보다 쉽다. 또한 찾고자 하는 근거리부터 원거리까지의 차선을 인식하기 위해서는 영상 전체를 처리하여 한꺼번에 차선을 인식하는 방법을 적용할 수도 있겠지만 근거리의 명확한 차선을 찾고 이를 기초로 하여 원거리 차선을 인식하는 방법이 처리 시간 면에서 많은 이점을 가질 수 있을 것이다.

다. 차선 인식 방법

차선의 영상은 자율이동로봇의 전방에 부착된 CCD 카메라를 통해 실시간으로 얻어진다. 로봇의 올바른 주행을 위해 실시간으로 얻어지는 영상으로부터 주행로의 차선을 정확하게 파악해야 한다. 그러나 차선의 영상은 주위 환경의 변화로 인하여 수시로 변하게 된다. 실제 주행하면서 차선을 추출하다보면 많은 어려움이 존재한다. 예를 들면 건물이나 도로변의 가로수 등의 그림자로 인하여 차선의 식별이 어려운 경우, 차선이 불연속적이기 때문에 이를 추출하지 못하는 경우, 주행로의 노화로 인해 차선이 지워져 식별이 어려운 경우 등이 있다. 그리고 주위 조도의 변화로 인해 카메라로 들어오는 주행로의 영상이 불완전하여 차선의 추출을 어렵게 하는 경우도 있다.

이외에도 주행을 하다보면 빛의 반사로 또는 이물질로 인해 차선이 그려져 있지 않은 경우에 주행을 계속해야 할지 혹은 자율이동로봇을 정지시켜야 할지 결정하는 것도 중요한 문제이다. 즉, 차선을 인식하지 못하고 주행을 계속하는 것은 많은 위험을 가지지만 차선을 인식 못하는 것이 어느 순간인 경우 로봇을 정지시키는 것 또한 불필요한 동작이기 때문이다. 차선 경계의 인식은 카메라로부터 입력된 영상으로부터 주행로의 차선을 전부 추출하는 것이 아니라 차선의 개략적인 윤곽만 추출할 수 있도록 몇 개의 지점만 추출한다. 이는 주행로의 형태가 몇 개의 점들만

추출하고도 충분히 그 형태를 추정될 수 있기 때문이다. 차선의 추출을 위해 영상화면상에서 수직으로 세 지점을 선택하여 그 각각의 지점으로부터 좌측과 우측방향으로 탐색하여 주행로를 이루는 4개의 점을 식별하도록 하였다.



그림 8. DSP에서 처리된 영상(좌)과 USB 카메라를 통한 영상(우)
(Fig. 8) The Image captured by DSP(L) & USB camera(R)

그리고 그림 8은 독립형(Stand alone)기반인 DSP로 처리한 영상과 USB카메라를 통하여 PC기반에서 처리한 영상이다. 독립형(Stand alone)기반에서는 알고리즘에 대한 디버깅이 불편하다. 이는 알고리즘을 수정할 경우 일일이 롬(ROM:Read Only Memory)에 프로그램을 써야 되기 때문에 디버깅 시간이 지체된다. 하지만 보다 빨리 알고리즘을 검증하는 방법은 PC 기반에서 USB 카메라를 비슷한 환경으로 꾸며 보다 빠른 시간 내에 알고리즘을 검증할 수 있어 도움을 줄 수 있다.

주행로 경계를 인식하는 방법을 간단히 살펴보면 다음과 같다. 여기서 y 좌표는 영상화면에서 수직 좌표이고, x 좌표는 수평좌표를 나타낸다. 다음은 자율이동로봇이 영상을 처리하는 과정을 순서대로 나타내었다.

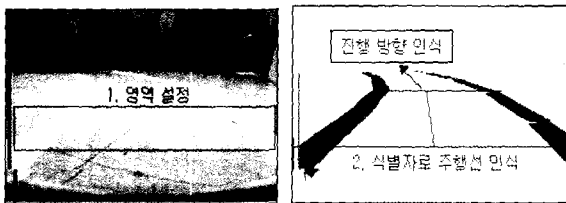


그림 9. 영역 설정과 식별자를 사용한 경계 인식
(Fig. 9) Setting a region & recognition of contour

- Step 1 : 영상 화면상의 2점의 y 좌표를 찾는다. 이 때 y 좌표의 위치가 영상화면에서 아래 부분에 위치할 수록 주행로 경계를 추출하는데 수월하지만 그 위치와 로봇과의 거리가 너무 가깝다면 원활한 자율주행을 하기 어렵게 된다.
- Step 2 : 탐색영역을 정한다. 바로 직전에 추출한 주행로 경계를 포함하는 6 개의 직사각형의 탐색 영역을 지정한다
- Step 3 : 차선 경계 식별자를 이용하여 주행로 경계를 인식한다.

이미 정해진 근거리 윈도우내에서 그림10의 에지추출 Operator를 사용한다. 영상처리시 에지를 추출하는 오퍼레이터로는 Sobel Operator를 사용했다. Sobel Operator는 수직 및 수평 에지성분을 찾는데 효과적이다. 로봇이 도로

주행 시 대부분 차선 중앙으로 주행하기 때문에 영상에 나타나는 전방도로상의 차선은 사선모양으로 나타난다. 따라서 (a)를 이용해서는 영상의 왼쪽에 나타나는 차선 에지를 찾는데 적용하고 (b)는 영상의 오른쪽에 나타나는 차선 에지를 찾는데 적용하도록 한다.

2	1	0
1	0	-1
0	-1	-2

0	1	2
-1	0	1
-2	-1	0

(a) (b)
그림 10. 소벨 오퍼레이터
(Fig. 10) Sobel Operator

그림 10을 적용하여 값을 계산한 후 에지임을 결정하기 위해서는 임계값과 비교해야 한다. 그런데 이 임계값을 고정 시켜 놓고 사용하게 되면 주위 환경에 따라, 특히 조명에 따라 에지 인식에 많은 차이점을 보일 수 있다. 따라서 도로 영상의 회색음영(Grayscale) 변화에 따라 임계값도 함께 변할 수 있도록 해야 한다.

도로영상에서 차선에 해당하는 부분은 0(검은색) ~ 255(흰색)의 회색음영(Grayscale)값 중 255에 가까운 영역에 해당할 것이다. 입력된 영상 중 흰색으로 만들어진 윈도우내의 회색음영(Grayscale)값을 히스토그램화한다. 이를 위해 본 연구에서는 로봇의 전방도로상의 일정부분을 주기적으로 읽어들이어 이를 히스토그램화한다

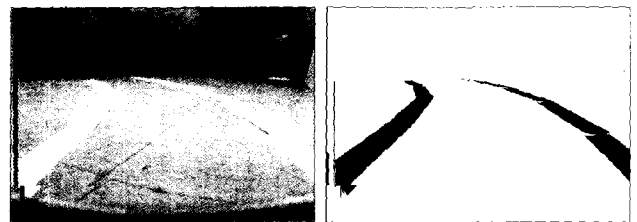


그림 11 DSP로 입력된 영상을 PC로 소벨에지 처리한 영상
(Fig. 11) Implementaion of Sobel operator

다음은 자율이동로봇이 차선을 인식하는 경계를 3가지의 방법으로 나누어 분류하였다

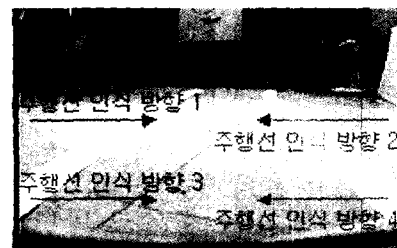


그림 12. 주행선 인식 방향
(Fig. 12) direction of recognized driving lane

- Case 1 : 양측의 경계 모두를 찾은 경우
왼쪽 경계점이나 오른쪽 경계점이 한 개 이상 발견된 경우에는 이전에 구해진 지수 평활화된 값

과 비교해 가장 가까운 지점을 주행로의 경계라고 결정한다. 좌측과 우측 경계의 지수 평활화된 값은 아래의 식(1)에 의해 계산된다.

$$X_{l_{avg}}(t+1) = a * X_{l_{avg}}(t) + (1-a) * X_l(t),$$

$$X_{r_{avg}}(t+1) = a * X_{r_{avg}}(t) + (1-a) * X_r(t) \quad (식1)$$

여기서, $0 \leq a \leq 1$,

$X_l(t), X_r(t)$: t시점에서의 좌측과 우측 경계의 x 좌표

1. 구해진 주행로 경계의 폭을 주어진 임계값과 비교하고 일정 범위 내에 있는지 검토
2. $d_1 < d_2$,
3. $c_1 \geq c_2$ 또는 $c_1 < c_2$

여기서, d_i : 각각의 y 좌표 내에서 도로 경계의 폭
 c_i : 각각의 y 좌표 내에서 추출한 도로 경계 중심의 x 좌표

Case 2 : 왼쪽이나 오른쪽 중 어느 한쪽의 경계를 인식하지 못한 경우

구하지 못한 경계에 대해서는 이전에 구한 $X_{l_{avg}}$ 나 $X_{r_{avg}}$ 의 정보와 도로 폭의 넓이의 자료를 이용하여 경계를 추정하도록 한다.

Case 3 : 양쪽의 경계를 모두 인식하지 못한 경우

불연속적인 도로의 여건이나 기타의 여러 가지 이유로 도로의 양측 경계 모두를 인식하지 못한 경우이다. 도로의 경계의 추정은 과거에 구한 자료들이 사용된다. 이 상황이 계속될수록 오차가 증가하게 되고, 일정 시간 이상인 경우는 주행의 계속여부를 결정한다.

차선을 인식 한 후 위의 3가지 경우에 따라 DC모터에 대해 제어를 할 수 있다. DC모터는 정지상태인 0부터 최고속인 20까지 21개의 계수로 제어가 된다. 차선을 검출하면 정지상태에서 계수를 올려 가속을 실시한다. Case 1 과 같이 양쪽 차선이 계속 검출되거나 Case 2 와 같이 한쪽 차선이 감지된 경우에는 가속을 멈추고 등속운동을 한다. 방향이 조향되면 약간의 감속이 이루어진다. Case 3 과 같이 차선을 인식하지 못한 경우에는 감속을 하게 되고 결국 정지하게 된다.

2. 방향제어

본 연구에서는 자율이동로봇의 이동각도를 계산하는데 수학적인 모델을 이용하여 조향각도를 구하는 방법을 사용하였다. 다음은 수학적인 모델을 이용하여 조향 각도를 구하는 방법에 대해 살펴본다. 본 연구에서는 카메라의 기

하학적인 모델링을 통하여 자율이동로봇의 이동각도를 계산하기 위해 몇 개의 좌표계를 정의하여 사용하였다. 그림 13.은 카메라 초점으로부터 전방 차선의 거리를 계산하기 위해 카메라에 관련 파라미터들을 $\theta = 89^\circ$, $f = 150\text{mm}$, $H = 220\text{mm}$ 로 설정하였다.

수학적인 모델을 이용하여 적절한 핸들의 조향각을 계산하기 위해서는 세 가지의 단계가 필요하다. 첫 번째는 화면 좌표계상에서의 위치를 계산한다. 이는 카메라로부터 얻어진 주행로의 영상을 처리하여 주행로의 차선을 추출한 다음 가고자 하는 위치를 결정함으로써 계산된다. 두 번째는 실제 로봇을 위치시키고자 하는 실 좌표계상에서의 위치의 결정이다. 화면 좌표계상에서의 위치를 월드 좌표계상에서의 위치로 변환시킴으로써 가능하다. 세 번째는 가고자 하는 위치가 정해진 후 그 위치까지 로봇을 이동시키기 위한 조향 각도의 결정이다. 이는 조향 각도를 결정하기 위해 필요한 여러 요인들 카메라의 높이와 초점거리, 로봇의 길이 등의 요소들을 이용하여 자율이동로봇의 방향각을 구한다.

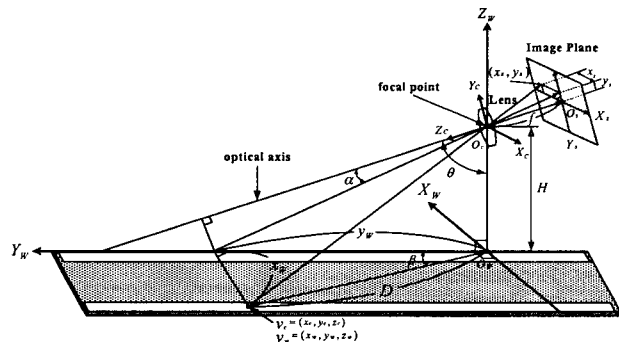


그림 13. 좌표계와 전방 차선과의 위치 관계
 (Fig. 13) Relation of location between driving lane and coordinated system

Step 1 : 화면좌표계상에서의 위치 계산

로봇의 진행 방향 각도를 구하기 위해 앞에서 구한 차선의 경계점들을 이용한다. 핸들 조향 각도를 구하는 목적은 핸들을 구동하여 로봇을 주행로의 중심에 위치시키도록 하는 것이다. 화면 좌표계상에서의 위치는 추출한 차선 중심점들의 화면상에서의 위치들 중 한 점을 선정하여 사용한다. 이러한 지점의 선정은 로봇이 원활한 주행을 하는데 중요한 관건이 된다. 너무 로봇에 가까운 지점을 선정하면 주행로의 형태를 반영하지 못하여 주행로 중심위치를로의 주행을 방해하기 쉬운 반면, 너무 로봇과 먼 지점을 선정하면 주행로의 곡률반경이 적은 경우에는 원활한 주행이 힘들게 된다. 이는 실험을 통해 적절한 위치를 선정하도록 한다.

Step 2 : 실 좌표 상에서의 위치 계산

자율이동로봇을 위치시키고자 하는 화면상의 좌표가 x_s, y_s 라 하고, 카메라의 높이(H)와 투사각(θ)을 각각 알 때 실제 로봇을 이동시키고자 하는 월드 좌표계 상에서의 위

치를 구할 수 있다. 이동시키고자 하는 지점은 주행로의 바닥에 위치함으로써 z_w 좌표가 0이 된다. 그래서 (식2)에서 처럼 화면좌표계를 이용하여 좌표를 구할 수 있다.

$$a = \tan^{-1} \frac{y_s}{f}$$

$$z_c = \frac{H}{\cos(\theta - a)} \quad (\text{식 } 2)$$

최종적으로 구하고자 하는 전방 로봇의 거리 D와 방향 β 는 각각 (식3)와 같다.

$$D = \sqrt{x_w^2 + y_w^2}$$

$$\beta = \tan^{-1} \left(\frac{x_w}{y_w} \right) \quad (\text{식 } 3)$$

영상내에 있는 차선의 거리를 측정하기 위해서는 카메라 캘리브레이션(Camera Calibration)을 하여야 한다. 수평도로 전방의 미리 정해진 거리를 로봇에 위치시키고 원하는 차선까지의 거리를 측정한 후 모델에 의해 계산된 거리와 일치하는지를 확인하는 방법을 사용하였다.

Step 3 : 방향 결정

실제 자율이동로봇의 정확한 회전궤적을 구하는데 그림 14의 방법을 이용하였다. 실제적으로 매우 구현하기 어렵기 때문에 앞바퀴의 조향각과 로봇의 길이만 고려한 모델을 구현하였다. 여기서는 주행 중인 로봇의 회전궤적에 대한 간단한 모델을 제시하고자 한다.

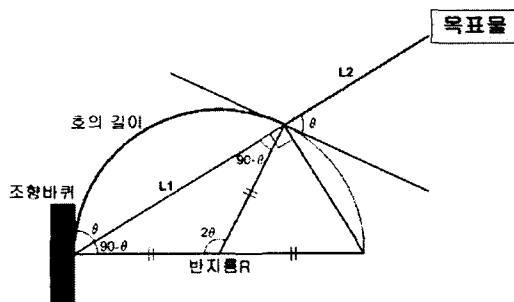


그림 14. 방향 결정 방법
(Fig. 14) determination of steering

목표물과의 거리 $L = L_1 + L_2$

$$L_1 = R \sin \theta \quad (\theta: \text{Radian})$$

$$L_2 = L - R_2 \sin \theta \quad (\text{식 } 4)$$

위와 같은 방법으로 결정된 방향에 따라 자율이동로봇의 RC모터의 계수가 변한다. RC모터의 계수는 로봇 중심의

방향을 0으로 놓았을 때 좌부터 우까지 $-120 \sim 0 \sim +120$ 으로 세밀하게 분류 하였는데, ± 30 씩 총 9개의 계수를 사용하여 방향을 전환한다.

IV. 실험 결과

본 로봇의 실험은 직선 트랙과 곡선 트랙 두 가지 환경에서 실시하였다. 직선 트랙은 자율이동로봇이 직선으로 된 차선을 인식하여 트랙을 이탈하지 않고 어느 정도의 속도로 자율 주행할 수 있는지를 알기 위해 이용하였고 곡선 트랙은 자율이동로봇의 곡선 주행 시 주행선 인식 오차와 속도를 알기 위하여 이용하였다. 트랙은 직경 50mm인 흰색 종이테이프를 이용하였는데, 직선 트랙은 실험 차선폭 50mm, 트랙 폭 250~300mm, 트랙 총길이 20m로 구축하였다. 곡선 트랙은 직선 트랙과 같은 환경에 곡률 반경 30~40m의 곡선 6개가 연속되게 총 길이 20m를 설치하였다.

직선 트랙 (20m)				곡선 트랙 (20m)			
	주행시간 (sec)	주행속도 (km/h)	절대오차 (cm)		주행시간 (sec)	주행속도 (km/h)	절대거리 (cm)
1회	2.17"	33.2	3.7	1회	3.18"	22.6	5.1
2회	2.09"	34.4	4.8	2회	3.31"	21.7	2.1
3회	2.15"	33.5	2.8	3회	3.27"	22.0	3.5
:				:			
30회	2.23"	32.3	3.1	30회	3.23"	22.3	4.9
평균	2.19"	32.9	3.6	평균	3.22"	22.3	4.1

표 2. 실험트랙에서의 주행결과
(Table. 2) The result of test track

실험용 직선 트랙에서 자율주행을 30회 실험해 본 결과 자율주행로봇은 최고 속도 35.1km/h, 평균 32.9km/h 속도로 주행을 하였다. 직선 트랙에서는 차선의 이탈 없이 비교적 정확하게 차선 검출을 하였는데, 차선을 인식한 후 주행 시 트랙의 가운데 기준점과 이동로봇 중심의 오차거리를 구해 본 결과 평균 3.6cm의 오차가 발생하였다. 실험용 곡선 트랙에서 자율주행을 30회 실험해 본 결과 총 30회 중 4회를 트랙에서 이탈하여 약 13%의 인식 오류를 보였는데, 이는 주변의 환경을 차선과 혼동하였기 때문이다. 곡선 트랙에서는 직선 트랙보다는 느린 평균 22.3km/h의 속도와 4.1cm의 오차거리를 자율주행 시 보여주었다.

PC 기반의 시스템과 비교를 위하여 자율주행로봇의 속도 방향제어부 기관을 대신해 방향제어 알고리즘을 구현한 노트북 컴퓨터를 시리얼로 연결한 시스템을 구성하여 실험

하였다. 직선과 곡선이 혼합된 40m의 타원형 트랙에서 각각 10회씩 실험한 결과 차선 인식 및 방향 제어는 비슷한 성능을 보였으며, 주행 속도는 독립형이 PC기반보다 빠른 속도를 보여주었다.

	독립형	PC기반
평균 주행시간(sec)	5.52	6.78
평균 주행속도(km/h)	26.0	21.2
평균 절대오차(cm)	3.7	3.8

Table 3. 독립형 시스템과 PC 기반 시스템의 결과 비교
(Table. 3) A comparison between standalone system and PC based system

V. 결 론

본 연구에서는 기존의 PC기반과 스테레오 비전을 사용한 주행형 이동로봇이 아닌 단일 카메라를 사용한 독립형기반의 시스템을 위하여 전방의 차선을 인식하여 주행 할 수 있는 소형 자율이동로봇 시스템에 대한 하나의 모델을 제시할 수 있었다. 또한 영상처리부, 구동부, 센서제어부, 전원부를 필요에 따라 독립, 시스템과 연계를 함으로써 각각의 시스템에서 지연 없이 동작할 수 있는 모듈화가 가능한 시스템을 제시하였다.

본 시스템은 단일카메라를 사용하여 차선을 인식하기 때문에 고속 주행에 필요한 응용분야에는 적용하기 적합하다. 하지만 장애물이나, 눈, 비, 안개 및 차선이 불분명한 도로 등과 같이 인식하기 좋지 않은 주행 환경에는 적용하기 어렵다는 단점이 있다. 이를 해결하기 위해서는 패턴인식 알고리즘을 사용해 불분명한 차선이라도 정확한 차선을 검출 할 수 있는 방법을 적용하는 것은 향후 연구과제로 남아있다.

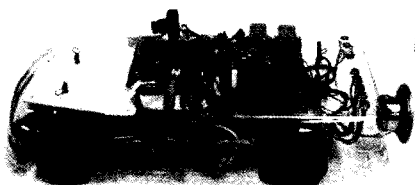


그림 15. 자율이동로봇
(Fig. 15) Autonomous mobile robot

접수일자 : 2003. 1. 03 수정완료 : 2003. 1. 20

참고 문헌

[1] 한민홍, 원윤재, "영상기반의 자율이동로봇", 고려대학교 석사 논문, 2001
 [2] 조준수, "차선이탈 경보장치를 위한 DSP 기반 영상처리장치 개발", 고려대학교 석사학위 논문, 2001.
 [3] 한민홍, 노광현, "자율주행차량개발현황", 자동차공학회지, 1998.
 [4] 한민홍, "자율주행차량기술의 현황", 대한기계학회지, 2001.
 [5] 문순환, "이동 운반체 자율주행 방법에 관한 연구", 고

려대학교 박사논문, 1996.
 [6] 김영학, "DGPS와 PCS를 이용한 시각장애자 보행보조 시스템 개발" 고려대학교 석사학위 논문, 2000.
 [7] 정승권,이외 4명,"무인차량의 강인한 조향제어 시스템 설계에 관한 연구", 제어 · 자동화 · 시스템공학 논문지 2002.9
 [8] Thorpe, Charles E., Vision & Navigation -The CMU Navlab, Boston : Kluwer Academic Publishers, U.S.A., 1990.
 [9] Wolfram Burgard, "The Interactive Museum Tour-Guide Robot", Proc. of the Fifteenth National Conference on Artificial Intelligence(AAAI-98), Madison, Wisconsin, 1998.
 [10] J. Matijevic, Mars Pathfinder Microrover - Implementing a Low Cost Planetary Mission Experiment", Jet Propulsion Laboratory - California Institute of Technology, 1997.
 [11] Stewart Moorehead, Reid Simmons, Dimitrios Apostolopoulos and William "Red" Whittaker "Autonomous Navigation Field Results of a Planetary Analog Robot in Antarctica", Carnegie Mellon University, The Robotics Institute, 1999.
 [12] Richard Hartley and Andrew Zisserman, "Multiple View Geometry in Computer Vision", Cambridge, p178~p182.



정 성 보(Sung-Bo Jung)
 准會員
 2000년 고려대학교 신문방송학과 졸업.
 2001년~현재 고려대학교 산업시스템정보공학과 석사과정.
 관심분야 : 이미지 프로세싱, 컴퓨터 비전, 로보틱스, 뉴럴 네트워크, 패턴 인식



이 경 복(Kyung-Bok Lee)
 准會員
 2001년 남서울대학교 산업공학과 졸업.
 2003년 고려대학교 산업공학 석사
 2003~현재 고려대학교 산업시스템정보공학과 박사과정.
 관심분야는 컴퓨터 비전, 로보틱스, 뉴럴 네트워크, 제어이론



장 동 식(Dong-Sik Jang)
 正會員
 1979년 고려대학교 산업공학과 졸업.
 1985년 텍사스 주립대 산업공학 석사
 1988년 텍사스 A&M 산업공학 박사
 1989~현재 고려대학교 산업시스템정보공학과 교수.
 관심분야는 컴퓨터 비전, 멀티미디어시스템, 로보틱스