

라인스캔 알고리즘을 이용한 AGV-UNO-CAR 설계 및 구현

장 준 영*, 인 치 호°

Design and Implementation of AGV-UNO-CAR Using a Line Scan Algorithm

JunYoeng Jang*, Chi-Ho Lin°

요 약

최근 AGV는 스마트 팩토리 등 산업현장에서 사용이 증가하고 있으며, 주로 유선유도 방식을 사용한다. 무선방식에 비해 저비용으로 구현 가능하며 AGV를 빠른 속도로 이동 시킬 수가 있다. 하지만 유선 유도 방식은 유도 라인을 작업장 바닥에 매립 또는 부착하는 방식으로 유도라인의 변경 및 유지보수가 어렵다는 단점이 존재한다. 본 논문에서는 유선 유도 방식의 단점을 해결하기 위해서 오픈소스 아두이노 컨트롤러와 저비용의 고속 비전 센서를 이용하여 주행 경로 및 바코드 인식을 통한 주행명령을 수행할 수 있는 라인 스캔 알고리즘을 제안한다. 16개의 바코드를 사용하여 AGV 운행경로의 재설정 및 변경이 가능하도록 하였다. 직접 제작한 AGV-UNO-CAR 로봇을 대상으로 한 실시간 운행 실험을 통하여 라인 스캔 알고리즘이 정상적으로 동작하는 것을 확인하였다.

Key Words : AGV, Barcode, CMU cam5 Pixy2, Linescan Algorithm, Otsu method

ABSTRACT

Recently, AGV is increasingly used in industrial sites as smart factories, and mainly uses a wired induction method. Compared to the wireless method, it can be implemented at a lower cost, and AGV can be moved at a high speed. However, the wired induction method has the disadvantage that it is difficult to change and maintain the induction line as it is a method of embedding or attaching the induction line to the floor of the workplace. In this paper, we propose a line-scan algorithm that can perform driving commands via driving path and barcode recognition using open-source Arduino controllers and low-cost high-speed vision sensors to address the shortcomings of wired induction methods. Sixteen bar codes were used to reset and alter AGV routes. A real-time driving experiment with a self-produced AGV-UNO-CAR robot confirmed that the line scan algorithm was operating normally.

1. 서 론

최근 정보 통신기술의 발전으로 인해 4차 산업 혁명이 많은 주목을 받고 있다. 이에 따라 빅데이터, 인

공지능, 사물인터넷 등 4차 산업혁명의 핵심 기술들에 관한 연구들이 활발하게 진행 중이다. 4차 산업 혁명의 핵심기술 못지않게 4차 산업혁명을 통해 변화되는 산업에도 많은 관심이 쏠리고 있다. 그중에서도 스마

* First Author : Semyung University, School of Computer, win4723@nate.com, 정희원

° Corresponding Author : Semyung University, School of Computer, ich410@semyung.ac.kr, 정희원

논문번호 : KICS202103-069-D-RN, Received March 24, 2021; Revised April 30, 2021; Accepted May 6, 2021

트 공장은 4차 산업 혁명으로 시작된 산업 중에서도 많은 사례와 함께 변화의 물결을 선도하고 있다.

스마트 공장을 구현하기 위한 여러 기술 중 산업계에서 주시하는 기술은 무인 운반차(AGV, Automated Guided Vehicle)기술이다.^[1,2] AGV는 1953년에 처음으로 사용된 기술이다. 이 당시 AGV는 트레일러를 끄는 형태로 사용되었으나 60년대 초반에는 다양한 공장과 창고에서 사용되었고, 70년대에는 스웨덴의 자동차회사 Volvo의 조립라인에서 사용되던 컨베이어를 AGV로 대체하면서 AGV의 활용이 산업 전반으로 확장되게 되었다. 현재의 AGV는 배터리로 움직이는 무인 반송차로 주어진 주행선에 따라 원하는 위치까지 이동하는 장비이다.^[3-6] AGV의 대표적인 예로는 미국 온라인 쇼핑 업체인 아마존의 키바 로봇을 들 수 있다. 아마존은 2012년에 키바 시스템을 인수한 후 100여 개의 물류 센터 중 10개의 물류센터에서 키바 로봇 30,000대를 운행 중이다. 아마존은 물류센터에서 키바 로봇을 이용하면서 주문 처리 비용을 최대 9억 달러(40%), 물류 센터 비용을 20% 절감하게 되었고, 물류 창고의 절반 이상의 공간을 더 효율적으로 사용할 수 있게 되었다.^[7] 우리나라의 경우에는 생산 총 인구는 감소하고 있으며, 그에 따른 인건비 증가로 인하여 기업에서 스마트 공장 구축으로 생산성을 향상시켜 원가를 절감하려는 목적으로 연구를 지속적으로 진행하고 있다.

AGV의 주행경로 유도 방식은 레이저 내비게이션, RFID 등을 이용한 무선 유도방식과 마그네틱, 광학, 전자기, 자기-자이로 유도 등의 유선 유도 방식이 있다.^[8-12] 무선 유도 방식은 가상의 경로를 사전에 생성시켜 위치 측정 센서들을 통하여 얻은 AGV의 현재 위치를 이용해서 목표 지점까지 유도하는 방식으로 유도 라인이 존재하지 않기 때문에 완전한 자율주행이 가능하며 주행 라인의 변경 및 유지보수가 편리하다. 하지만 무선으로 유도가 이루어지기 때문에 전파의 외란, 빛의 굴절과 같은 신호의 왜곡 현상뿐만 아니라 장애물로 인하여 측정의 오차가 생길 수 있다. 또한, 작업장의 환경과 AGV에 설치되는 송/수신기의 특성에 따라 정밀도의 차이가 크다는 단점이 있다. 현재 산업현장에서는 주로 유선유도 방식을 사용하는데 추적방식이 저렴하게 구현 가능하며, AGV를 빠른 속도로 이동시킬 수 있는 방법이기 때문이다. 하지만, 유선 유도 방식은 유도 라인을 작업장 바닥에 매립 또는 부착하는 방식으로 잦은 유도라인의 변경 및 유지보수가 어렵다는 단점이 존재하여 이를 극복하기 위한 다양한 연구가 진행되고 있다.^[13-16]

AGV의 위치를 파악하는 기술은 이미 레이저, 마그네틱, 관성 등을 이용한 여러 가지 방법들이 사용 중에 있으며, 위치 인식을 위한 다양한 센서를 개발 중이다. 레이저 스캐너를 사용하는 방식은 반사판을 설치하여 레이아웃의 변경이 쉽고, 바닥에 매설 작업이 필요 없어 클린룸 환경에서도 사용이 가능하다. 하지만, 통로가 좁고 주변에 반사되는 물체가 많은 경우 측정의 오차가 심해 주행이 어려운 구간이 존재한다. 자기 테이프를 사용하는 방식은 자기테이프를 바닥에 매설한 후 마커센서를 사용하여 매설된 궤적을 따라가는 방식이다. 주행 궤도를 매설하여 주변 환경에 영향을 받지 않아 안정적인 운행이 가능하다. 하지만 반도체, 화학제품 등의 클린룸 환경에서는 주행궤도 매설자체가 어렵기 때문에 사용이 어렵다. 이 방법들은 안정적이지만 고비용이고, 유지 및 변환이 어렵다는 단점을 가지고 있다. 이러한 단점들의 해결을 위해 새로운 위치 파악 기술에 대한 연구가 주목받고 있다.^[17-20] 기존의 AGV의 위치파악 기술의 단점을 해결하기 위해 QR코드 같은 2차원의 이미지를 이용하거나 RFID 태그, 카메라를 이용하는 등의 다양한 연구가 진행되고 있다. 카메라를 사용한 비전기반의 위치 인식 기술은 크게 환경내 특징점을 기반한 위치인식 기술과 인공마커 기반의 위치인식 기술로 구분할 수 있다. 특징점을 기반한 위치 인식기술은 다양한 환경에 적용하기 위해 추가적인 작업은 필요 없지만, 특징점 인식을 위한 여러 단계의 복잡한 수식을 필요로 한다. 또한 특징점이 적은 작업환경에서는 적용이 어려운 문제점이 있다. 인공마커 기반의 위치인식 기술은 AR이나 QR마커 같은 마커들을 사용하여 위치를 인식하는 기술로 특징점이 적은 환경에서도 약속된 패턴을 인식하기 때문에 빠르고 정확한 위치인식이 가능하다.^[21-23]

본 논문에서는 유선 유도 방식의 단점을 해결하기 위해서 CMUcam5 Pixy2라는 저비용의 고속 비전 센서와 바코드를 이용한 라인 스캔 알고리즘을 제안한다. 제안된 방법은 저비용의 고속 비전 센서를 이용한 카메라 영상을 사용한 주행 경로 및 바코드 인식을 통한 주행 명령을 동시에 수행 할 수 있도록 한다. 제작한 AGV-UNO CAR 로봇에 제안한 라인 스캔 알고리즘을 적용하여 경로를 인식하여 주행하는 기능과 바코드를 확인하여 주행환경에 따른 주행 명령을 변경할 수 있도록 하였다.

II. AGV-UNO-CAR 주행 시스템

그림 1은 논문에서 사용한 AGV-UNO-CAR 로봇의 전체 구성도를 나타내고 있다. 메인 보드로 ATmega328을 사용하였고 비전 센서와는 ISP포트를 통하여 연동을 하였다. 비전센서는 연속적으로 촬영이 가능하고, 특징점이 적어 빠른 응답속도를 가지는 CMUcam5 pixy2를 사용하였다. 또한 비전 센서 보드에 서보모터 2개를 사용하여 상하좌우 제어가 가능하도록 하였다. AGV-UNO-CAR의 구동 모터로는 서보형 DC모터를 좌측 2개, 우측 2개를 사용하였고, 4개의 모터 구동을 위한 모터드라이버는 TB6612FMG를 사용하였다. 전원은 독립전원을 사용하였다.

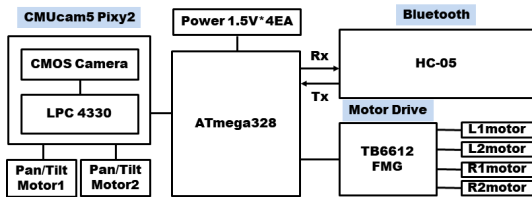


그림 1. AGV-UNO-CAR 전체 구성
Fig. 1. AGV-UNO-CAR Overall Configuration

2.1 Main Controller

그림 2는 논문에서 사용한 AGV-UNO-CAR 주행 시스템의 메인 컨트롤러를 나타내고 있다. 프로세서는 ATmega328을 사용하였으며 블루투스 칩드와 모터 구동 칩드가 내장되어 있다. 메인컨트롤러는 16Mhz 클럭으로 동작하며, 2개의 1.2A 모터를 구동할 수 있는 모터드라이브, 무선 업로드 및 ISP 기능을 가지고 있으며, 14개의 디지털 입·출력 핀과 6개의 아날로그 입력 핀을 가지고 있다.

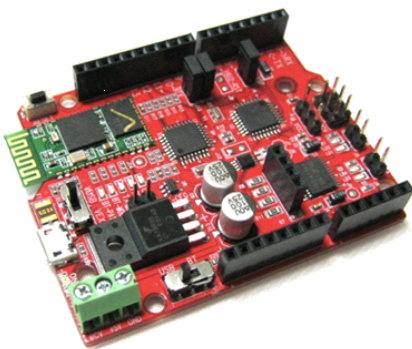


그림 2. AGV-UNO-CAR 메인 컨트롤러
Fig. 2. AGV-UNO-CAR Main Controller

2.2 CMUcam5 Pixy2

그림 3은 라인 및 바코드를 확인하기 위해 사용한 CMUcam5 Pixy2 비전센서를 나타내고 있다. 비전센서는 작고 응답속도가 빠르며 사용하기 편리하다. 저비용으로 바로 사용 가능한 비전 시스템으로 학습을 통한 사물 인식 기능, 아두이노 및 라즈베리파이 등을 위한 라이브러리 제공, C/C++ 및 Python 호환이 가능하다. 또한 SPI, I2C, UART, USB 또는 아날로그/디지털 출력, 윈도우/MacOS 및 리눅스 호환이 가능한 환경설정 유틸리티를 제공한다. 주요성능으로는 다음과 같다.

- Processor : NXP LPC4330, 204 MHz, dual core
- Image sensor : Aptina MT9M114, 1296 × 976
- Lens field-of-view : 75 degrees horizontal, 47 degrees vertical
- Ram : 264K bytes
- Flash memory : 2M bytes

그림 4는 AGV-UNO-CAR 로봇 제작에 사용된 전체 회로도를 나타내고 있다.

그림 5는 AGV-UNO-CAR 주행 시스템의 주행 실험을 위해서 입체 교차로 및 평면 교차로의 다양한 주행 환경에서 주행명령을 수행할 수 있도록 가상의 실험

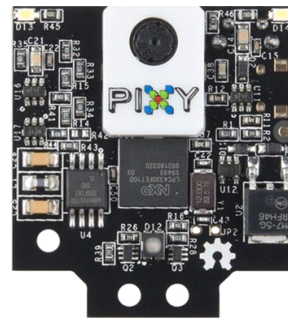


그림 3. CMUcam5 Pixy2
Fig. 3. CMUcam5 Pixy2

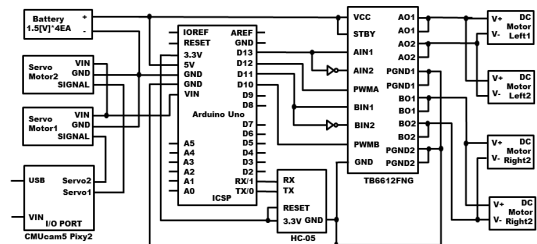


그림 4. AGV-UNO-CAR 회로도
Fig. 4. AGV-UNO-CAR Circuit

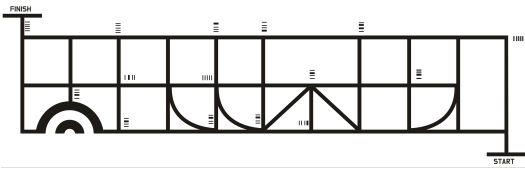


그림 5. AGV-UNO-CAR 가상의 주행 시스템 실험 환경
Fig. 5. AGV-UNO-CAR Virtual Driving System Experimental Environment

현환경을 구성하였다. AGV-UNO-CAR 로봇은 흰색 배경에 검은색 라인으로 이루어진 주행 경로를 카메라로 촬영하며 라인 추적 알고리즘에 따라 라인을 인식하고 분기점에서는 바코드에 따라 주행명령을 수행하도록 설계하였다.

III. 라인추적 및 스캔 알고리즘

그림 6은 논문에서 사용한 AGV-UNO-CAR 동작 알고리즘을 나타내고 있다. 라인과 바코드를 확인하여 주행 명령을 수행하는 라인 스캔 알고리즘을 적용하였다.

라인 인식을 위한 방법으로 Otsu method를 사용하였다. 영상에서 이미지를 추출 후 이미지에서 정보를 얻어내기 위해서는 이미지가 담고 있는 개체들을 분리하는 작업을 해야 한다. 개체를 분리하는 대표적인 방법 중 하나가 이진화이고, Otsu method는 다양한 이진화 방법들 중 하나이다. Otsu method는 영상의 히스토그램 형태가 bimodal 이라고 가정하였을 때, valley 값을 찾아서 그 값을 임계값으로 사용하는 것으로, 클래스를 기반으로 이진화를 수행한다. Otsu

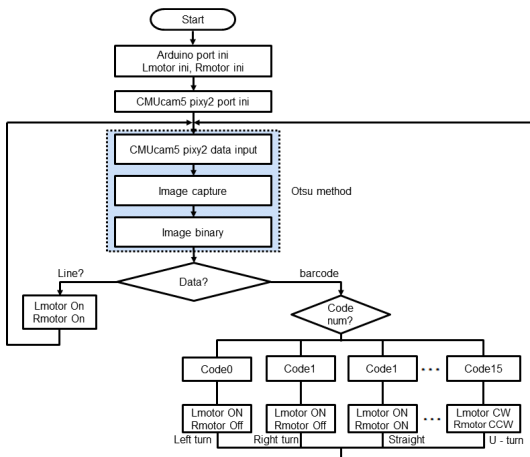


그림 6. AGV-UNO-CAR 동작 알고리즘
Fig. 6. AGV-UNO-CAR Operational Algorithm

method의 이진화 방법은 임계값을 기준으로 영상 픽셀들을 두 클래스로 분류 했을 때 두 클래스간의 intra-class variance를 최소화 하거나 inter-class variance를 최대화 하는 임계값을 찾는 이진화 방법이다. 입력 영상에서 임계값보다 어두운 픽셀들의 비율을 ω_0 밝은 픽셀의 비율을 ω_1 , 밝기 평균을 μ_0, μ_1 , 임계값을 t , 클래스의 분산을 σ_0^2, σ_1^2 이라 했을 때 intra-class 분산과 inter-class 분산은 식(1)과 (2)처럼 계산할 수 있다.

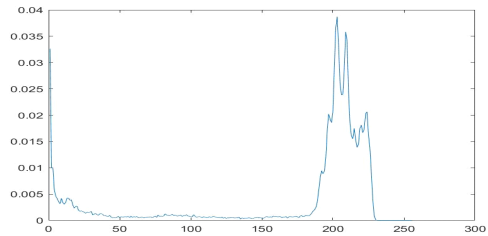
$$\sigma_w^2(t) = \omega_0(t)\sigma_0^2(t) + \omega_1(t)\sigma_1^2(t) \tag{1}$$

$$\begin{aligned} \sigma_b^2(t) &= \sigma^2 - \sigma_w^2(t) = \omega_0(\mu_0 - \mu_T)^2 + \omega_1(\mu_1 - \mu_T)^2 \\ &= \omega_0(t)\omega_1(t)[\mu_0(t) - \mu_1(t)]^2 \end{aligned} \tag{2}$$

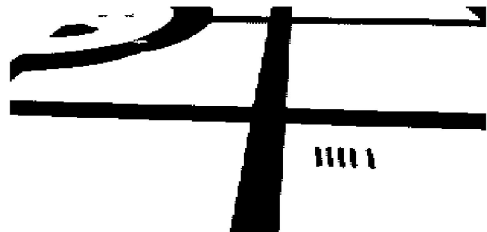
식 (1)을 최소화시키는 것은 식 (2)를 최대화 시키는 것과 동일하며 둘 중 어느 기준을 사용해도 같지만 계산상 식 (2)를 최대화 시키는 것이 효율적이어서 본



(a) Original image



(b) Histogram



(c) Otsu method result screen

그림 7. 교차로 우회전 바코드
Fig. 7. Intersection and Right Barcode

논문에서는 식 (2)를 사용하여 프로그램 하였다. Otsu method를 사용한 알고리즘은 matlab으로 확인하였다.

그림 7(a)는 분기가 4개인 라인과 우회전 바코드의 원본 영상이며, 그림 7(b)는 원본영상에서 Otsu method를 사용해 matlab을 통해 얻은 히스토그램을 나타내고 있다. 또한 7(c)는 Otsu method를 사용해 이진화 시킨 결과 이미지를 나타내고 있다.

주행 명령을 내리기 위해 사용한 바코드는 굵거나 가는 검은 막대와 흰 막대의 조합으로 문자, 숫자 또는 특수 기호를 부호화한 단순한 광학 장치이다. 문자나 숫자를 이진수 0과 1로 바꾸고 이들을 조합해 정보로 쓰는데, 이들은 심벌로지라고 하는 바코드 언어의 규칙에 따라 만들어 진다. 인쇄된 코드는 인식장치의 빛 반사를 이용해 데이터를 재생시키고 재생된 데이터를 모아 업무를 처리하도록 전송하고 있다. 바코드는 일정한 규칙에 따라 만들어져 바의 두께와 여백의 비율에 따라 여러 종류의 코드 체계가 완성 된다.^[24] 그림 8에서는 CMUcam5 pixy2 카메라에서 사용할 수 있는 바코드를 나타내고 있으며, 총 16개의 바코드

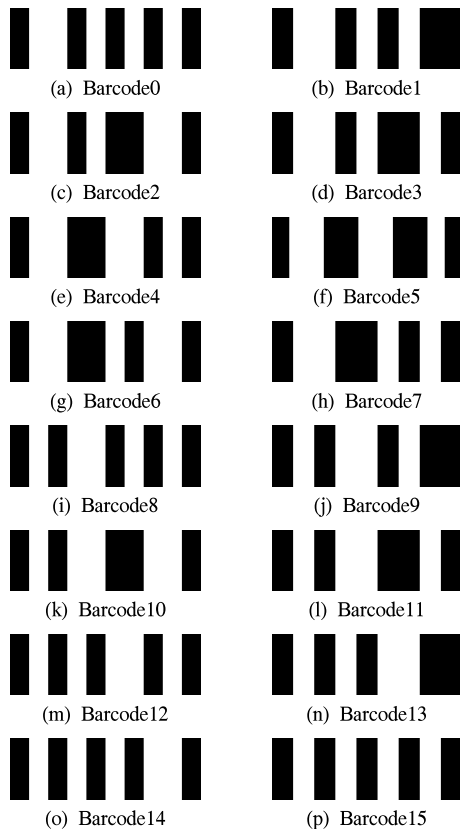


그림 8. 16개 바코드
Fig. 8. 16 Barcode

```
#include <Pixy2.h>
#include <PIDLoop.h>
#include <ZumoMotors.h>
#include <ZumoBuzzer.h>
#define ZUMO_FAST 200
#define ZUMO_SLOW 150
#define X_CENTER (pixy.frameWidth/2)
Pixy2 pixy;
ZumoMotors motors;
ZumoBuzzer buzzer;
PIDLoop headingLoop(5000, 0, 0, false);
void setup()
{ Serial.begin(115200);
  Serial.print("Starting...\n");
  motors.setLeftSpeed(0);
  motors.setRightSpeed(0);
  pixy.init();
  pixy.setLamp(1, 1);
  pixy.changeProg("line");
  pixy.setServos(500, 1000); }
void loop()
{ int8_t res;
  int32_t error;
  int left, right;
  char buf[96];
  res = pixy.line.getMainFeatures();
  if (res<=0) {
    motors.setLeftSpeed(0);
    motors.setRightSpeed(0);
    buzzer.playFrequency(500, 50, 15);
    Serial.print("stop");
    Serial.println(res);
    return; }
  if (res&LINE_VECTOR)
  { pixy.line.vectors->print();
    headingLoop.update(error);
    left = headingLoop.m_command;
    right = -headingLoop.m_command;
    if (pixy.line.vectors->m_y0 >
      pixy.line.vectors->m_y1) { if
      (pixy.line.vectors->m_flags&LINE_FLAG_INTERSECTION_PRESENT)
        { left += ZUMO_SLOW;
          right += ZUMO_SLOW; }
        else { left += ZUMO_FAST;
          right += ZUMO_FAST; } }
        else { left -= ZUMO_SLOW;
          right -= ZUMO_SLOW; }
        motors.setLeftSpeed(left);
        motors.setRightSpeed(right); }
  if (res&LINE_INTERSECTION)
  { buzzer.playFrequency(1000, 100, 15);
    pixy.line.intersections->print(); }
  if (res&LINE_BARCODE)
  { buzzer.playFrequency(2000, 100, 15);
    pixy.line.barcodes->print();
    if (pixy.line.barcodes->m_code==0)
      pixy.line.setNextTurn(90);//right
    else if (pixy.line.barcodes->m_code==1)
      pixy.line.setNextTurn(-90)//left
    else if (pixy.line.barcodes->m_code==2)
      pixy.line.setNextTurn(0); stright
    else if (pixy.line.barcodes->m_code==3)
      pixy.line.setNextTurn(-180);//u-turn
    else if (pixy.line.barcodes->m_code==15)
      pixy.line.setNextTurn(-90); }
    //m_code 값은 0~15까지 사용
    //m_code 값을 사용하여 명령 변경
    //pixy.line.setNextTurn(degree);
    //degree 값을 변경하여 모터회전 각도 변경}
```

그림 9. AGV-UNO-CAR 동작소스 코드
Fig. 9. AGV-UNO-CAR Operating Source Code

를 사용할 수 있다. 바코드 중 임의적으로 선택하여 주행 명령 중 좌회전, 우회전, 직진, 유턴에 해당하는 바코드로 사용하였다. 또한 다양한 작업환경에서 필요한 여러 가지 명령들을 수행 할 수 있다.

그림 9는 AGV-UNO-CAR 주행 시스템의 동작 소스코드를 나타내고 있다. m_code0~m_code15 값을 사용하여 다양한 주행환경에서 각종 명령을 수행할 수 있다.

IV. 실험 및 결과

그림 10은 제안한 라인 스캔 알고리즘을 적용 및 실험을 위한 AGV-UNO-CAR 로봇의 구성을 나타내고 있으며 크게 CMUcam5 pixy2, Pan/Tilt servo motor1*2ea, Power 1.5[V]*4ea, JARDUINO, DC Motor*4ea, Bluetooth HC-05로 구성되어 있다.

그림 11은 AGV-UNO-CAR 주행 시스템의 주행 실험을 위해서 가상의 실험환경을 제작하여 주행실험 환경을 보여주고 있다.

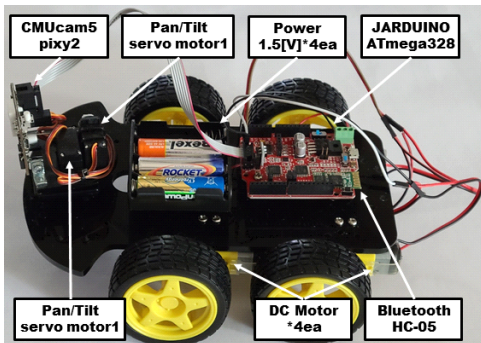


그림 10. AGV-UNO-CAR
Fig. 10. AGV-UNO-CAR

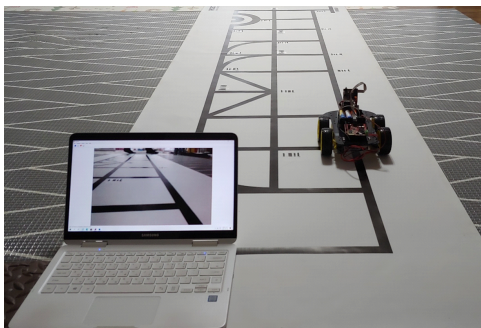


그림 11. AGV-UNO-CAR 주행 시스템 실험 환경
Fig. 11. AGV-UNO-CAR Driving System Experimental Environment

라인 스캔 알고리즘을 사용하기 위해 추적하기 쉬운 기본 라인을 정해 LINE_VECTOR라는 변수에 값을 할당을 하였다. LINE_VECTOR는 로봇이 현재 추적하고 있는 선을 의미한다. getAllFeatures()라는 함수를 사용하여 라인 스캔 알고리즘에서 감지하는 모든 선, 교차점, 바코드 값을 반환하도록 프로그램 하였다. 추적 알고리즘에서 감지된 교차점에서의 좌회전, 우회전, 직진, 유턴 등의 주행 명령은 바코드를 인식하여 처리하도록 하였다. 그림 12는 라인 스캔 알고리즘에서 사용할 총 16개의 바코드를 나타내고 있으며 모니터링 프로그램을 이용하여 인식한 실험 결과 화면을 보여주고 있다.

라인 스캔 알고리즘에서는 프레임 상에 새로운 라인이 나타나면 각 라인을 추적하여 각 라인이 후속 프레임에서 이동하는 위치를 결정하므로 로봇이 추종하는 라인은 따로 지정하지 않는 한 그대로 유지되도록 하였다. 각 프레임에서 벡터가 시작되고 끝나는 위치를 알려줘서 해당 정보를 사용하여 로봇의 동작을 결정하도록 프로그램 하였다. 예를 들어 벡터가 프레임

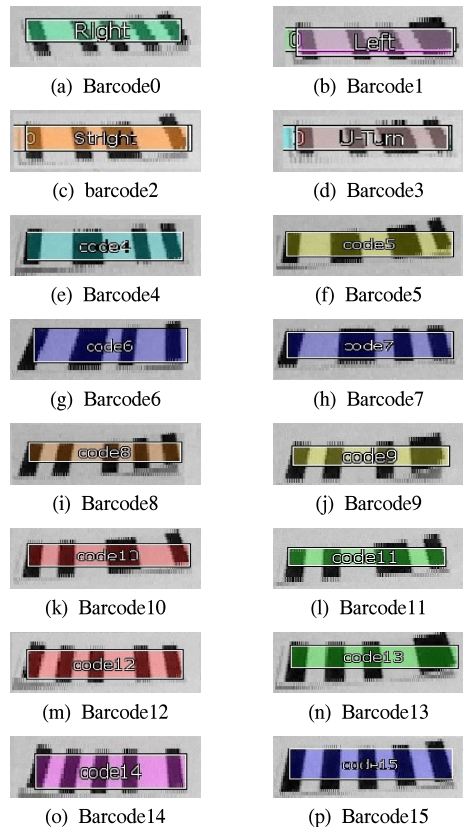


그림 12. 바코드 인식 실험결과
Fig. 12. Experimental Result that Recognized Barcode

에서 오른쪽으로 향하면 로봇이 오른쪽으로 회전하기 시작하고 벡터가 왼쪽으로 향하면 로봇이 왼쪽으로 회전하기 시작하도록 프로그램 하였다. 스캔 알고리즘에서는 교차점이 확인되면 전방 교차점 및 교차점의 분기 개수, 바코드, 분기 각도를 감지하여 어떤 분기로 라인 추적을 할지 결정하게 되면 새로운 분기를 벡터 라인으로 만들고 새 벡터의 방향으로 움직이도록 프로그램 하였다. 그림 13은 라인과 바코드 값이 제대로 인식되는지 모니터링 프로그램을 통하여 얻어진 실험 결과 값들을 보여주고 있다. 그림 13(a), 13(b), 13(c), 13(d)는 교차로에서 좌회전, 우회전, 직진, 유턴 명령의 실험 결과를 나타내고 있다. 13(e), 13(f)는 삼거리에서의 실험 결과, 13(g)는 곡선이 포함된 사거리, 13(h)는 각도가 있는 오거리의 실험 결과 값을 각각 나타내고 있다. 입체 교차로 및 평면 교차로의 다양한 주행환경에서 주행명령 및 다양한 명령을 수행할 수가 있다.

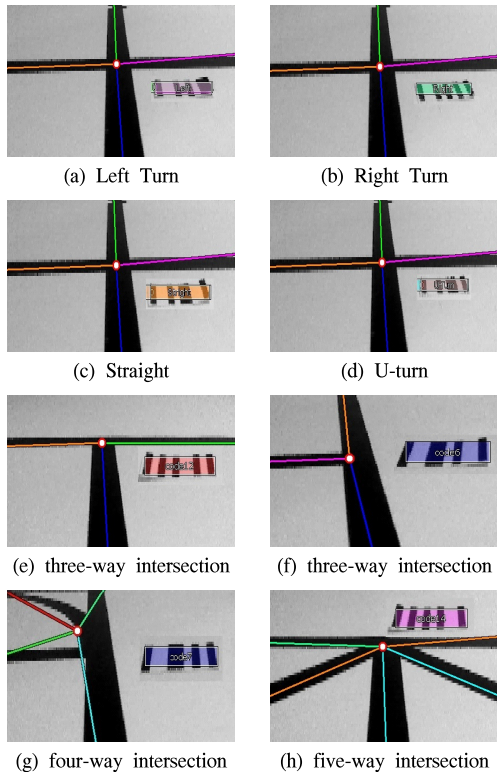


그림 13. 주행명령을 인식한 라인스캔 알고리즘 결과
Fig. 13. AGV-UNO-CAR Line Scan Algorithm Result that Recognized Driving Command

V. 결 론

본 논문에서는 유도 라인을 작업장 바닥에 매립 또는 부착하여 유도라인의 변경 및 유지보수가 어려운 유선 유도 방식의 단점을 해결하기 위해서 CMUcam5 Pixy2라는 저비용의 고속 비전 센서와 바코드를 이용한 라인 스캔 제어 알고리즘을 제안하였다. 먼저 저비용의 고속 비전 센서를 이용한 카메라로 영상을 입력받아 이미지를 전송받았다. 전송된 이미지에 Otsu method를 이용한 이진화를 적용하여 인식된 라인과 바코드에 라인 스캔 알고리즘에 사용하여 경로 인식 및 주행 명령을 수행하도록 하였다. 제작된 AGV-UNO-CAR 로봇을 이용한 주행 실험을 통해 제안된 라인 스캔 알고리즘의 타당성을 입증하였다.

기존의 AGV는 설치된 유도라인 및 분기점에서 정해져 있는 라인으로만 주행명령을 수행 하였다. 새로운 작업환경 및 추가적인 명령을 수행하기 위해서는 유도 라인을 추가 및 재설치를 해야 하고 소스코드를 변경하여야 하였지만 제안한 라인 스캔 알고리즘을 적용한 AGV-UNO-CAR 로봇은 라인 및 바코드에 따라 주행명령을 수행하기 때문에 별도의 유도라인의 재설치 없이 바코드만을 추가함으로써 새로운 주행명령 및 추가 명령을 수행할 수가 있다. 입체 및 평면 교차로 등의 다양한 주행환경에서 라인 변경 및 훼손에 따른 유도라인의 추가적인 설치 없이 바코드 변경만으로 다양한 주행명령을 수행할 수가 있어 무인운반차 주행 시스템의 설치비 원가 절감 및 설비 구축 시간을 단축시킬 수 있는 장점을 갖는다. 표 1은 기존의 AGV 방법과 본 논문에서 제안한 AGV 방법의 기능을 비교한 것이다.

향후 AGV-UNO-CAR 주행 시스템을 산업현장에 적용하여 정확성 및 타당성을 입증하는 과제가 남아 있다.

표 1. 기능 비교
Table. 1. Feature Comparison

Item	conventional method	proposed method
Terminal change Line (Add Line)	○	×
cost of installation	expensive	cheap
facility build time	long	short

References

- [1] I. A. Vis, "Survey of research in the design and control of automated guided vehicle systems," *Eur. J. Oper. Res.*, vol. 170, no. 3, pp. 677-709, 2006.
- [2] T. Le-Anh, "A review of design and control of automated guided vehicle system," *Eur. J. Oper. Res.*, vol. 171, pp. 1-23, 2006.
- [3] L. Qui, W. J. Hsu, S. Y. Huang, and H. Wang, "Scheduling and routing algorithms of AGVs: A survey," *Int. J. Prod. Res.*, vol. 40, no. 3, pp. 745-760, 2002.
- [4] Y. C. Ho, "A dynamic-zone strategy for vehicle collision prevention and load balancing in an AGV system with a single-loop guide path," *Comput. Ind.*, vol. 42, no. 2-3, pp. 159-176, 2002.
- [5] H. Hwang and S. H. Kim, "Development of operation Rules for automated guided vehicle systems in heterarchical manufacturing system," *J. KIIE*, vol. 23, no. 2, pp. 343-357, 1997.
- [6] R. L. Moorthy, W. H. Guan, N. W. Cheong, and T. C. Piaw, "Cyclic deadlock prediction and avoidance for zone-controlled AGV system," *Int. J. Prod. Econ.*, vol. 83, no. 3, pp. 24-30, 2003.
- [7] S. H. Kim and H. G. Lee, "Implementation of pattern recognition algorithm using line scan camera for recognition of path and location of AGV," *J. Korea Ind. Info. Syst.*, vol. 23, no. 1, pp. 13-21, 2018.
- [8] J. Borenstein, "The OmniMate: A guidewire and beacon-free AGV for highly reconfigurable applications," *Int. J. Prod. Res.*, vol. 38, no. 9, pp. 1993-2010, 2000.
- [9] C. Y. Chan, "A system review of magnetic sensing system for ground vehicle control and guidance," California PATH Research Report, UCB-ITS-PRR-2002-20, 2002.
- [10] M. J. Caruso, T. Bratland, C. H. Smith, and R. Schneider, "A new perspective on magnetic field sensing," *Sensor Mag.*, vol. 15, no. 12, pp. 34-46, 1998.
- [11] C. Y. Chan and H. T. Tan, "Evaluation of magnetic as a position reference system for ground vehicle guidance and control," California PATH Research Report, UCB-ITS-PRR-2003-8, Mar. 2003.
- [12] Y. J. Ryoo, E. S. Kim, and Y. C. Lim, "Intelligent positioning system for magnetic sensor based autonomous vehicle," *SCIS & ISIS*, 2004.
- [13] T. Kawano, M. Hara, and M. Sugisaka, "Generating target path for tracing a line before missing the traced line of dead angle of camera," *IEEE Conf. Int. Joint Conf.*, pp. 5286-5289, 2006.
- [14] C. Guosheng, Z. Deguai, X. Gang, and Z. Honbing, "Design and implementation of intelligent tracing algorithm based on machine vision," *IEEE Conf. Intell. Info. Technol. Appl.*, pp. 697-700, 2009.
- [15] G. Beccari, S. Caselli, F. Zanichelli, and A. Calafiore, "Vision-based line tracking and navigation in structured environments," *IEEE Conf. Computational Intell. in Robotics and Automat.*, pp. 406-411, 1997.
- [16] Z. G. Man, W. H. Ye, P. Zhao, P. H. Lou, and T. J. Wu, "Research on RFID and visionbased AGV navigation," *Advanced Mater. Res.*, vol. 136, pp. 298-302, 2010.
- [17] B. H. Choi, B. S. Kim, and E. T. Kim, "Location estimation and obstacle tracking using laser scanner for indoor mobile robots," *J. KIIS*, vol. 21, no. 3, pp. 329-334, 2011.
- [18] S. W. Heo and T. H. Park, "Localization system of AGVs using laser scanner and marker sensor," *J. Inst. Contr., Robotics and Syst.*, vol. 23, no. 10, pp. 866-872, 2017.
- [19] S. Kamewak and S. Uemura, "A magnetic guidance method for automated guided vehicles," *IEEE Trans. Magnetics*, vol. 23, no. 5, pp. 2416-2418, 1987.
- [20] D. Tao and J. J. Huo, "AGV laser positioning sensor systems research and development," Wuhan University of Technology, 2006.
- [21] T. G. Kim and N. Y. Ko, "Comparison and analysis of methods for localization of a

mobile robot,” *J. Advanced Info. Technol. and Convergence*, vol. 11, no. 1, pp. 79-89, 2013.

[22] J. Li, Z. Wu, J. Z. hang, and W. Chen, “Research of AGV positioning based on the two-dimensional code recognition method,” *2015 Int. Conf. Logistics, Informatics and Serv. Sci.*, pp. 1-6, 2015.

[23] B. K. Kim and R. R. Jung, “Control of an omni-directional mobile robot based on camera image,” *J. KIIS*, vol. 24, no. 1, pp. 84-89, 2014.

[24] S. H. Jang and J. P. Jeong, “Design and implementation of OpenCV - based inventory management system to build small and medium enterprise smart factory,” *J. IIBC*, vol. 19, no. 1, pp. 161-170, 2019.

장 준 영 (JunYeong Jang)



2003년 : 한라대학교 전기전자 제어공학부 졸업(학사)
2010년 : 한라대학교 대학원 공학석사
2014년~2017년 : 세명대학교 박사수료
2018년~현재 : 한라대학교 겸임교수

<관심분야> AGV, AI, Open Source Platform, Control theory

[ORCID:0000-0002-4565-4935]

인 치 호 (Chi-Ho Lin)



1985년 : 한양대학교 공과대학 전자공학과 졸업(공학사)
1987년 : 한양대학교 대학원(공학석사)
1996년 : 한양대학교 대학원(공학박사)
1992년~현재 : 세명대학교 컴퓨터학부 교수

<관심분야> Soc CAD, ASIC 설계, CAD 알고리즘, AI SoC 설계, 지능형 컴퓨팅 및 임베디드 시스템, 지능형 교통시스템 설계

[ORCID:0000-0003-0073-4779]