



저작자표시-비영리-동일조건변경허락 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.
- 이차적 저작물을 작성할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



동일조건변경허락. 귀하가 이 저작물을 개작, 변형 또는 가공했을 경우에는, 이 저작물과 동일한 이용허락조건하에서만 배포할 수 있습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권으로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

공 학 석 사 학 위 논 문

퍼지 규칙을 이용한 비전 및 무선 센서 네트워크
기반의 이동로봇의 자율 주행 및 위치 인식



2008년 8월

부 경 대 학 교 대 학 원

전 자 공 학 과

허 준 영

공 학 석 사 학 위 논 문

퍼지 규칙을 이용한 비전 및 무선 센서 네트워크 기반의 이동로봇의 자율 주행 및 위치 인식

지도교수 강 근 택

이 논문을 공학석사 학위논문으로 제출함.

2008년 8월

부 경 대 학 교 대 학 원

전 자 공 학 과

허 준 영

허준영의 공학석사 학위논문을 인준함.

2008년 8월 27일



주	심	공학박사	이 원 창	인
위	원	공학박사	강 근 택	인
위	원	공학박사	주 문 갑	인

목 차

I. 서론	1
II. 시스템 구성	4
21 시스템 개요	4
22 Robot	5
23 Node	6
24 User	7
III. 자율 주행 알고리즘	8
31 주행 알고리즘	8
32 위치 추정 알고리즘	22
IV. 실험 및 실험 결과	26
41 실험 환경	26
42 주행 및 장애물 회피 결과	27
43 노드 인식 및 위치 추정 결과	29
V. 결 론	32
VI. 참고 문헌	33

Navigation and Localization of Mobile Robot Based on Vision and Sensor Network Using Fuzzy Rules

Jun Young Heo

Department of Electronic Engineering, The Graduate School
Pukyong National University

Abstract

Algorithms of path planning and obstacle avoidance are essential to autonomous mobile robots that are working in unknown environments in the real time. This paper presents a new navigation algorithm of an autonomous mobile robot with vision, IR sensors and Zigbee sensor network using fuzzy rules. Temporary targets are set up by distance variation method and then the algorithms of trajectory planning, obstacle avoidance and localization are designed using fuzzy rules. An autonomous mobile robot with single vision and IR sensors, Zigbee sensor network is built up for experiments. We also show that the proposed algorithms are working well in complex unknown environments.

I. 서론

로봇 시스템은 과거 초창기 로봇에서부터 고속 연산과 센서 기술의 발달로 인해 군사, 우주 탐사, 의료 등 사회 전반에 걸쳐 널리 사용되고 있다. 초창기 로봇은 과학자들이 여러 가지 자동장치를 만드는 것으로 시작하였다. 기원 후 1세기 그리스의 과학자 헤론이 만든 ‘동전을 넣으면 물이 나오는 기계’, 1773년 스위스의 자케드로즈가 만든 태엽으로 움직이는 ‘글 베끼는 인형’, 1737년 프랑스의 보오킨슨이 만든 ‘톱니바퀴를 사용한 오리’ 등이 있었다. 그 후 체코의 극작가 카렐 차페크가 발표한 희곡에서 처음으로 로봇이란 말이 나오게 되었고, 아지작 아시모프는 로봇법칙인 로봇공학 3원칙을 만들게 되었다. 이것을 발판으로 오늘날까지 제너럴모터즈사의 최초 산업용 로봇, 한국 최초 보행 로봇인 ‘센토’, 일본 혼다사의 2족 보행 로봇인 ‘아시모’ 등 많은 로봇들이 개발되고 있다.

오늘날 로봇 시스템은 지능형 서비스 로봇 형태로 단순 반복적인 작업 수행에서 벗어나 자율성을 부여하여 동적인 환경에서 능동적인 활용이 가능하도록 설계되고 있다. 동적인 환경에서 로봇이 자율적으로 이동하기 위해서는 현재 로봇이 위치한 지역의 부정확하고 불완전한 정보로부터 결론을 유도해 낼 수 있어야 하는데, 이러한 자율 주행을 위해 퍼지 로직, 신경망, 유전자 알고리즘 등 지능시스템을 이용한 제어방식이 많이 연구되어지고 있다[1][2][3].

로봇이 자율 주행을 하기 위해서는 환경 인식, 경로 설정 및 장애물 회피, 위치 추정 과정이 필수적이다. 환경 인식, 경로 설정 및 장애물 회피를 위한 방법으로는 사용이 간단하면서도 원거리 측정이 가능한 초음파 센서,

우수한 정밀도를 가지는 레이저, 그리고 구조가 간단하면서도 근거리 측정이나 특정 물체의 감지가 가능한 IR 센서 등이 널리 사용되어져 왔으며, 최근에는 마이크로프로세서의 성능향상으로 CCD 카메라 사용이 가능해졌고 이로 인해 보다 많은 주변정보를 얻을 수 있게 되었다[4][6]. 그리고 위치 추정 방법으로는 로봇의 이동 바퀴에 회전수를 측정할 수 있는 encoder를 장착하여 이동거리를 계산하는 방법, 자이로 센서나 가속도 센서를 이용하는 방법, 일정한 패턴이나 특징점 등 landmark를 이용하는 방법, RFID 또는 GPS를 이용한 방법 등이 있다. 그리고 추가적으로 최근에는 환경 인식 및 위치 추정 방법으로 U-IT839 국가 전략으로 지정된 3대 인프라 중 하나인 USN(Ubiquitous Sensor Network)이 정책 및 산업적인 지원으로 활발히 연구됨으로서 실시간적인 주변 환경 인식을 위해 USN의 핵심 기술 중 하나인 무선 센서 네트워크를 통한 방법도 최근에 널리 사용되어지고 있다[9]. 이러한 무선 센서 네트워크는 센서 모듈과 네트워크 모듈로 이루어진다. 무선 센서 네트워크 프로토콜에서 물리 계층과 MAC(Medium Access Control) 계층은 IEEE 802.15.4 표준이 채택되고 네트워크 계층과 응용 계층은 Zigbee 표준이 채택된다[8]. Zigbee를 이용하여 구성된 센서 노드들은 유기적으로 동작하는 하나의 네트워크를 만들고 온도, 가스와 같은 주변 환경의 변화된 정보를 수집, 전송 및 처리하거나 전송된 정보를 노드 사이에서 라우팅 하는 역할을 수행한다. 또한 Zigbee를 이용한 통신기술은 다른 무선 통신기술과 비교해 전력 소모가 적고 생산비용이 저렴하며, 10 ~ 20미터 안팎의 근거리 통신에서 간단한 신호를 주고받는데 매우 유용한 기술이다.

본 논문에서는 미지의 환경에서 로봇의 자율주행을 위한 방법으로 Vision과 IR센서를 이용하였고, 현재 큰 관심을 받고 있는 Zigbee를 이용하여 효과적인 주행 방법을 제안하였다. 1대의 CCD 카메라와 IR 센서들을

이용하여 주행에 필요한 장애물 회피와 경로 설정을 하였으며, 모터 encoder와 Zigbee의 RSSI값을 이용하여 위치 추정을 하였다. 그리고 Zigbee를 이용해 하나의 센서 네트워크를 만들고 주변 환경 정보를 수집하고 전송하도록 하였고, 이러한 센서들로 얻어진 정보를 바탕으로 외부 잡음에 강인한 퍼지 제어기를 사용해서 목표지점까지 이동하도록 하였다[7].



II. 시스템 구성

2.1 시스템 개요

구현된 로봇은 화재 진압을 목적으로 그 크기를 줄여서 설계하였다. 시스템의 전체적인 다이어그램은 그림 1과 같으며, 크게 Robot, Node, User로 구분 지을 수 있다. 본 논문에서는 Robot 블록과 Node 블록 사이의 행동에 대한 효과적인 알고리즘에 대해서만 얘기하며, User 블록은 화재 진압 로봇으로서의 완성도를 높이기 위해 추후 요구되어지는 부분이다.

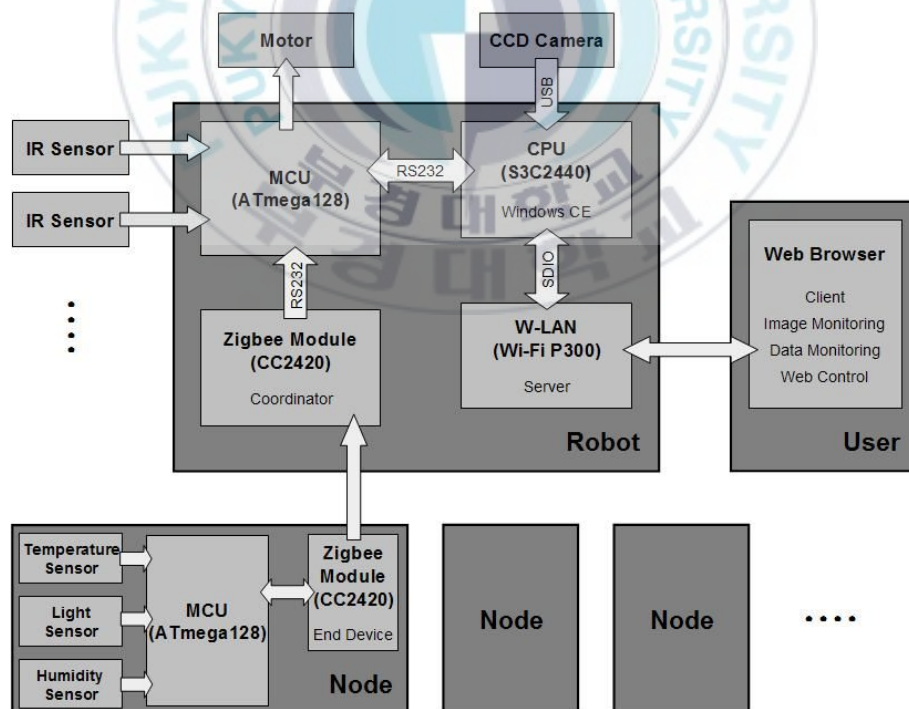


그림 1. 시스템 구성도

2.2 Robot

로봇은 노드들과 무선 센서 네트워크를 구성한 후 End Device로 동작하는 노드들로부터 화재 발생 여부를 판단할 수 있는 환경 정보에 대한 데이터를 입력받는다. 만약 어떤 노드에서 화재가 발생하였다면 Vision과 IR 센서를 이용해 경로 설정 및 장애물 회피를 하면서 주행을 하게 된다.

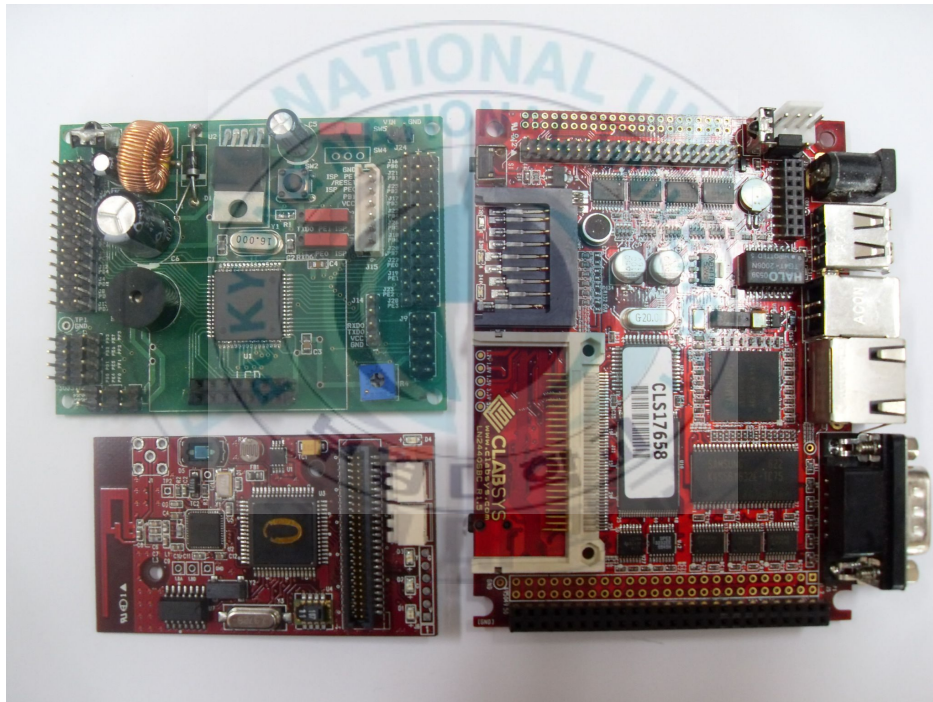


그림 2. S3C2440, ATmega128, CC2420 Board

그림 2의 왼쪽 위쪽에서부터 시계방향으로 ATmega128 보드, S3C2440 보드, CC2420 보드이다. 로봇의 구동에는 스텝모터가 사용되고, 장애물 인식을 위해서는 웹캠과 거리 측정이 가능한 IR 센서인 PSD 센서를 사용한

다. 이러한 모터와 IR 센서의 제어와 Coordinator로 동작하는 CC2420 보드로부터 네트워크에서 들어오는 센싱 데이터 등을 로봇의 핵심 부분인 S3C2440 보드로 전송해준다. S3C2440 보드는 ARM9 계열의 삼성 CPU S3C2440가 탑재된 SLABSYS사의 LN2440SBC 보드를 사용하였고, 이 보드에 Windows CE를 포팅해서 전체적인 제어 알고리즘을 구현하였다.

2.2 Node

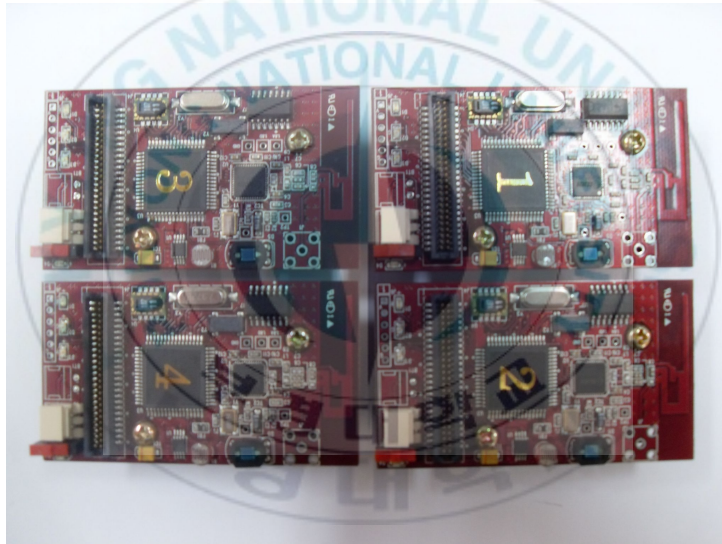


그림 3. Zigbee Module

그림 3은 한백전자의 Zigbex로 IEEE 802.15.4의 표준 그룹에 포함된 Zigbee Module이다. Module에는 온도, 조도, 습도를 센싱하기 위한 각각의 센서들이 있다. 또한 센서 제어, 통신 제어, 네트워크 구성을 위해 ATmega128 MCU도 내장 되어 있다. 그리고 CC2420과 ATmega128 사이에 통신 및 다른 CC2420 보드와의 무선 통신을 위해 ATmega128에

TinyOS를 올려서 보다 효과적으로 제어하도록 하였다. 통신은 IEEE 802.15.4에 기반하고 있으며, 이러한 IEEE 802.15.4 Frame Format은 다음 그림 4와 같다.

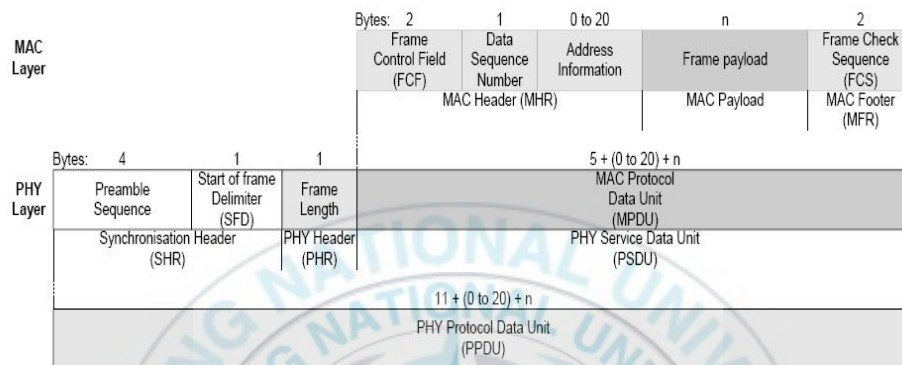


그림 4. IEEE 802.15.4 Frame Format

그리고 각각의 노드들은 Coordinator로 동작하는 로봇과 네트워크를 구성하면서 센싱된 데이터를 로봇에게 보내주게 된다.

2.2 User

화재 진압 로봇으로 사용되기 위해서는 사용자가 직접 로봇을 컨트롤하고 언제, 어디서든지 로봇이 위치한 주변 환경 정보를 알 필요가 있다. 이를 위해 로봇에는 SDIO 방식인 Socket사의 P300과 같은 Wireless-LAN Card를 S3C2440 보드에 붙여서 서버로 동작시켜야 한다. 이 후 사용자는 Client로 웹 브라우저를 통해 로봇에 들어오는 영상 및 환경 센싱 정보 등을 Monitoring하며, 직접 로봇도 Control 할 수 있게 된다.

Ⅲ. 자율 주행 알고리즘

3.1 주행 알고리즘

3.1.1 장애물 인식을 위한 영상처리

CCD 센서에서 얻어진 영상을 그림 5와 같은 순서를 거쳐서 장애물 인식 및 임시목표 설정을 위해 필요한 정보를 구한다.

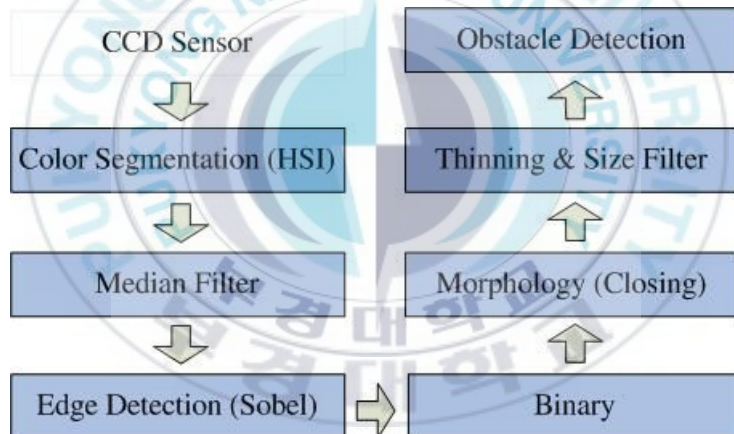


그림 5. 영상 처리 순서

먼저 입력된 컬러영상을 HSI 색상 평면으로 나누고 원 영상과 가장 유사한 명도성분(Intensity)을 나타내는 영상을 구한다. 기본적인 RGB 포맷을 사용할 수도 있지만 이는 빛의 밝기에 따라 색이 민감하게 반응해버리기 때문에 영상처리할 이미지로서 적절하지 못하다. 따라서 색상과 명도 성분이 분리되어 있는 HSI 포맷을 사용하였다.

그리고 디지털 영상에 발생하는 랜덤한 잡음들을 제거하기 위해서 잡음 필터 기법을 사용하여야 한다. 이러한 잡음을 제거하기 위하여 자주 쓰이는 방법이 저주파 통과 필터이다. 그러나 저주파 통과 필터링은 가우시안 잡음을 제거하는데 적합하지만 임펄스 잡음을 제거하기 위해서는 적합하지 않다. 임펄스 잡음에 의해 왜곡된 영상은 0 또는 255와 같은 뚜렷하게 잘못된 밝기값을 가진 많은 화소들을 갖게 된다. 이러한 신호들에 저주파 통과 필터링 기법을 적용하면 그 신호의 이웃한 화소들이 영향을 받는다. 그리고 필터 마스크가 클수록 이 신호에 영향을 받는 화소의 수는 많아진다. 따라서 임펄스 잡음을 제거하기 위한 효과적인 방법으로는 메디언 필터링 (Median Filtering) 기법을 사용하였다. 메디언 필터링은 비선형 영역처리 기법으로 저주파 통과 필터에 비하여 강한 에지를 보존하고 기존의 에지들을 보다 상세하게 보존할 수 있다. 필터링 방법은 한 영상의 화소들에 대하여 임의 크기의 윈도우를 슬라이딩하면서 오름차순으로 윈도우 안에 있는 화소들을 정렬하고 그 중간값을 선택하여 출력 영상을 채운다. 이렇게 함으로써 영상 자체는 흐릿해지나 노이즈는 없앨 수 있게 된다. 만약 그림 6의 첫 번째 그림과 같은 3×3의 이미지가 있다고 하자. 이 경우 화소값이 큰 순서대로 정렬하면 16, 9, 9, 8, 7, 5, 4, 3, 1이 되고, 이 중에서 중간값인 7을 선택하게 되므로 결과는 그림 6의 두 번째 그림과 같게 된다.

1	16	5
8	9	7
3	4	9

➡

1	16	5
8	7	7
3	4	9

그림 6. 메디언 필터링

그리고 영상 안에서 영역의 경계를 나타내야 하는데 이러한 Edge를 얻는 방법에는 Derivate, Gradient, Laplacian 연산자 등과 공간필터인 Roberts, Prewitt, Sobel 연산자를 사용하는 방법이 있다. 본 논문에서는 이러한 공간필터 중의 하나인 Sobel 연산자를 사용하였다. 이미지 위의 임의 좌표 (x , y) 위치에서의 Gradient는 식 (1)과 식 (2)와 같이 구해진다.

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (1)$$

$$\nabla f = mag(\nabla f) = \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{\frac{1}{2}} \quad (2)$$

그리고 구해진 Edge 영상을 적은 메모리량과 빠른 연산을 위해 이진영상으로 변환시켰다. 이진영상은 농담 값이 0 또는 1로서, 256개의 농담값 또는 칼라를 갖는 화상에 비해 효과적인 처리가 가능하다. 이러한 이진영상을 만들기 위해서는 임계값 처리 과정이 필요하고, 양자화 과정을 통해 얻어진 각 화소는 주어진 범위의 정수값을 갖게 된다. 임계값 범위 내에 있는 화소값은 1, 그 밖에 있는 것은 0으로 이진화하며 식 (3)과 같이 정의되어진다. 이 때 $F[i,j]$ 는 영상의 i행, j열 화소의 농담값을 나타내고, T는 임계값을 나타낸다.

$$F_T[i,j] = \begin{cases} 1 & \text{if } F[i,j] \geq T \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

이렇게 구해진 임계값에 의한 이진영상은 불필요한 조명에 의한 반사부분이 노이즈로 나타나게 되는데 이를 제거하기 위해서 모폴로지 기법을 사용하였다. 이는 물체와 배경 상이에서 물체의 모양이나 윤곽을 매끄럽게 하거나 축소 및 확장을 통해 물체를 보다 명확히 구별해낼 수 있다. 기본적인 처리 기법으로는 Dilation과 Erosion이 있는데, Dilation은 물체의 외곽을 두껍게 하거나 확장시키는 작용을 하므로 물체 내부의 빈 공간을 메우거나 가까운 거리의 끊어진 부분을 연결시켜주는 역할을 한다. Erosion은 물체의 외곽을 가늘게 하거나 축소시키는 작용을 하므로 아주 작은 물체를 제거하여 배경으로 만들어주는 역할을 한다. Opening은 먼저 Erosion을 수행하여 미세 잡음을 제거한 다음 Dilation을 수행하여 원 영상의 크기를 유지하도록 하는 것이고, Closing은 먼저 Dilation을 수행하여 물체를 확장하여 작은 구멍들을 채운 다음 Erosion을 수행하여 원 영상의 크기가 유지되도록 하는 것이다. Dilation 연산은 3×3 윈도우 내의 요소들의 모든 값이 0이면 그대로 0을, 하나라도 0이 아니면 255를 반환하며, Dilation Mask는 그림 7의 첫 번째 그림과 같다. 그리고 Erosion 연산은 3×3 윈도우 내의 요소들의 모든 값이 255이면 그대로, 하나라도 255가 아니면 0을 반환하며, Erosion Mask는 그림 7의 두 번째 그림과 같이 된다. 본 논문에서는 흰 고립점을 제거하기 위해 Closing 연산을 사용하였다. 즉 어떤 화소 가까이에 하나라도 0이 있으면 그 화소를 0으로 처리하는 Erosion 연산을 사용한 후에 어떤 화소 근처에 하나라도 255가 있으면 그 화소를 255로 처리하는 Dilation 연산을 사용하였다.

0	0	0
0	0	0
0	0	0

255	255	255
255	255	255
255	255	255

그림 7. Dilation Mask와 Erosion Mask

그리고 이렇게 구해진 이미지의 골격선으로 표시하면 정보의 양을 최대한 압축할 수 있을 뿐만 아니라 물체의 길이, 위치 등 특징을 추출하는 것도 용이해진다. 따라서 이를 위해 대상 물체의 표면을 조금씩 벗겨내어 최종적으로 두께 1을 가지도록 하는 세션화 과정을 거쳤고, 라벨링 및 크기 필터링을 통해 최종적으로 임시목표 설정을 위한 영상을 얻었다[5].

3.1.2 장애물과의 거리 및 각도 계산

구해진 장애물이 인식된 영상에서 픽셀들 사이의 변화율을 계산하여 급격하게 변화율의 차이가 나는 부분을 찾아 이를 장애물의 모서리로 간주하고, 이런 급격한 변화를 보이는 모서리들 중에서 변화율이 가장 큰 모서리를 임시목표로 정하였다. 임시목표가 정해지면 로봇과의 거리를 계산해야 하는데 이를 위해 구해진 영상의 정보와 IR 센서를 이용하였다. 두 개의 카메라에서 각각 얻어진 두 영상을 비교해 보다 정확한 거리를 산출할 수 있는 스테레오비전 방법도 있지만 본 논문에서는 하나의 카메라로 그림 8과 같은 구조를 이용하여 거리를 측정하였다.

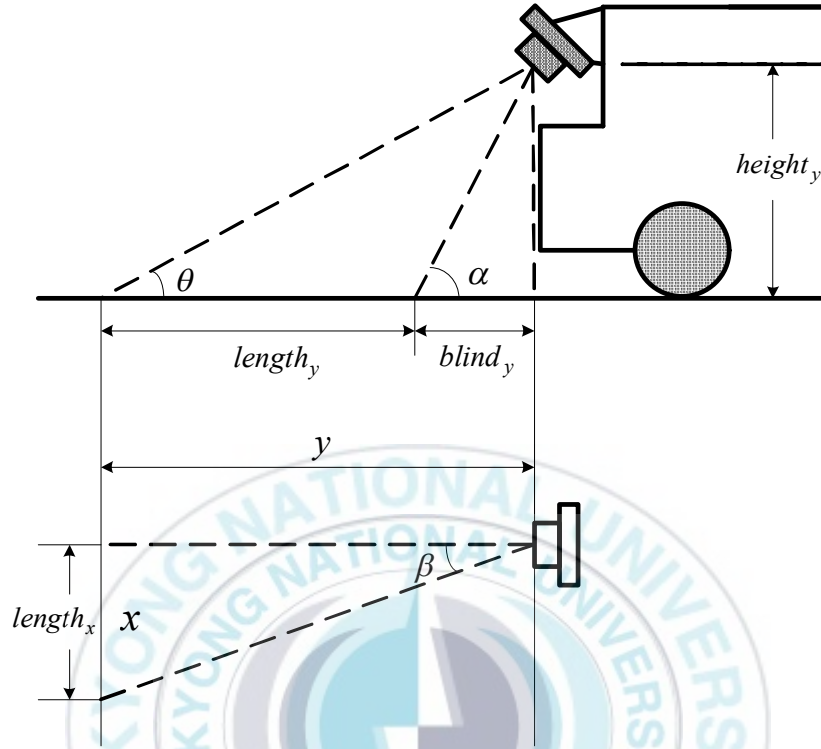


그림 8. 로봇의 측면도와 상면도

바닥에서 카메라까지의 높이인 $height_y$, 카메라 앞부분에 존재하는 사각의 거리인 $blind_y$, 카메라로 포착 가능한 최장거리인 $length_y + blind_y$, 카메라의 좌우 폭인 $length_x$ 는 실제 측정가능하고, 이 4개의 측정값과 식 (4), 식 (5), 식 (6)을 이용하여 다음의 세 가지 각도를 구할 수 있다.

$$\alpha = \tan^{-1}\left(\frac{height_y}{blind_y}\right) \quad (4)$$

$$\theta = \tan^{-1}\left(\frac{height_y}{blind_y + length_y}\right) \quad (5)$$

$$\beta = 90^\circ - \tan^{-1}\left(\frac{blind_y + length_y}{length_x}\right) \quad (6)$$

그리고 이 3가지의 각도와 식 (7), 식 (8)을 이용하여 임시목표점의 실제 공간좌표를 구해낼 수 있다.

$$y = \frac{height_y}{\tan\left(\theta + \frac{pixel_y(\alpha - \theta)}{screen_y}\right)} + blind_y \quad (7)$$

$$x = \tan\left(\frac{\beta(1 - 2pixel_x)}{screen_x}\right) \times y \quad (8)$$

$pixel_x$, $pixel_y$ 는 화면상의 x , y 좌표를 나타내며, $screen_x$, $screen_y$ 는 실제 입력된 영상의 가로, 세로 픽셀의 크기이다. 그리고 실험에 사용된 변수들의 값은 표 1과 같다.

표 1. 실험에 사용된 변수 값

$length_x$	276	$height_y$	24	$screen_x$	320
$length_y$	848	$blind_y$	30	$screen_y$	240

비전을 이용해 계속 주행을 하다가 임시목표와의 거리가 $blind_y$ 와 가까워지면, 즉 y 가 40보다 작아지는 시점부터는 IR 센서를 이용해 주행 및 장애물 회피를 한다. IR 센서는 그림 9와 같이 22.5° 간격으로 9개의 센서를 부착하였다. 큰 원은 로봇을 의미하고, 작은 원은 IR 센서를 의미한다.

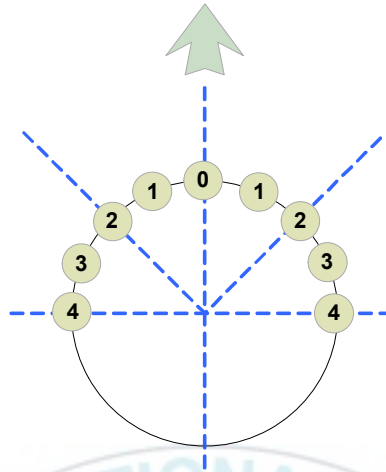


그림 9. IR 센서 배치도

IR 센서로부터 거리값이 자동으로 계산되어지기 때문에 9개 중 거리값이 가장 작은 것이 비전에서 정한 임시목표점이라 간주하고 이 IR 센서의 인덱스 번호에 22.5를 곱해서 로봇과 임시목표점과의 틀어진 각도를 구해낼 수가 있다.

3.1.3 퍼지 제어기

장애물 인식, 로봇과의 거리 및 각도 그리고 임시목표가 정해지면 주행을 위한 기본 요소들이 갖춰졌으며, 임시목표까지 진행하도록 하는 제어 동작이 요구되게 된다. 이러한 제어에는 종래의 로봇의 기구학과 동력학을 바탕으로 하는 PID 제어기와 퍼지 제어기, 신경망, 유전자 알고리즘 등의 지능시스템을 적용한 방법 등 여러 제어 방법이 있는데, 본 논문에서는 외부 잡음에 강인하고, 현재 수행하려는 작업처럼 수학적 모델로 표현하기 힘들거나 비선형 시스템의 제어에 적합한 퍼지 제어기를 사용하여 경로 생성을 하였다. 사용된 퍼지 제어기의 설계방법은 숙련자의 언어적 표현으로

부터 제어기를 설계하는 방식을 택하였고, 설계방식은 그림 10과 같다.

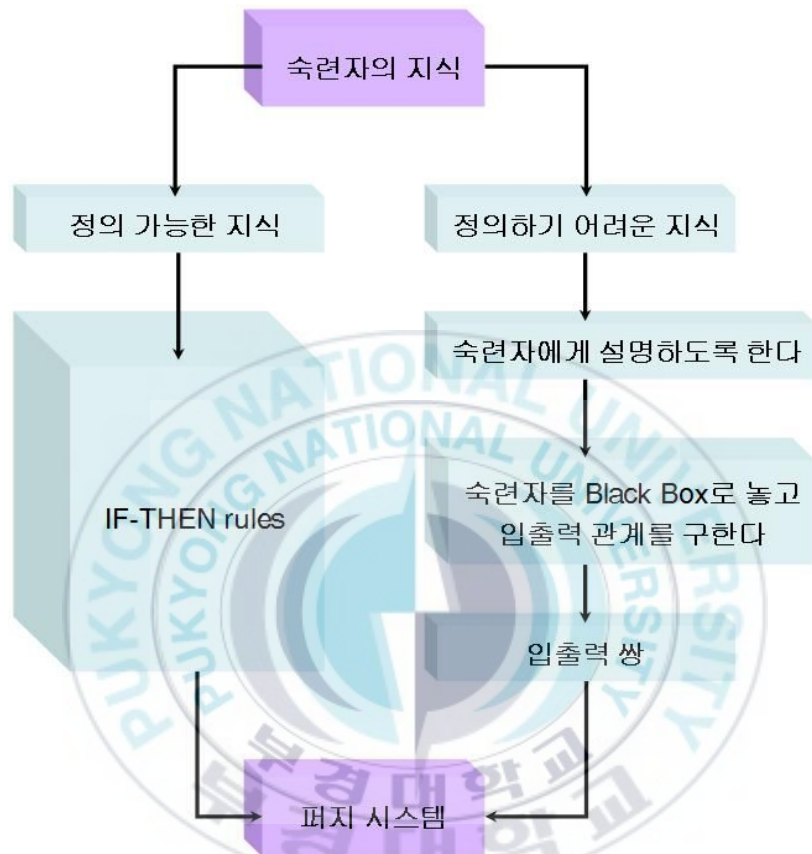


그림 10. 숙련자의 지식을 이용한 퍼지 시스템의 설계

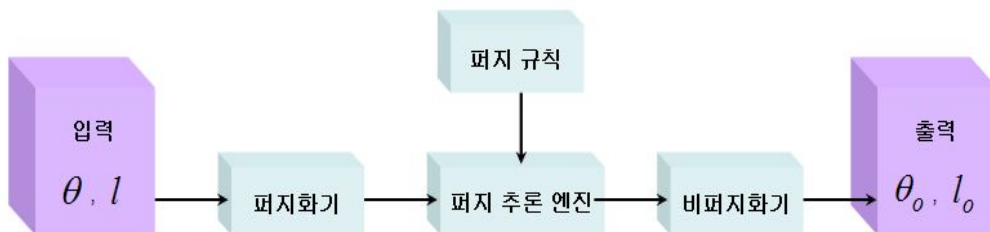


그림 11. 퍼지 시스템의 기본 구조

본 논문에서 사용된 제어기는 그림 11과 같이 제어기의 입력으로는 임시 목표와 로봇과의 거리(l), 임시목표와 로봇의 진행방향과의 차이(θ)를 사용하였고, 출력은 회전각(θ_o)과 이동해야 할 거리(l_o)로 설정하였다. 퍼지화기에 사용된 입력 소속 함수로는 계산이 간단한 삼각파를 사용하였고, 출력 소속 함수는 Singleton을 하용하여 연산량을 최소화하였다. 추론 방법으로는 Mamdani의 Min-Max 연산을 사용하였으며, 식 (9), 식 (10)과 같다.

$$\mu_{bi'}(y) = \text{supt}[(\mu_{A'}(x), \mu_{A' \rightarrow B_i}(x, y))] \quad (9)$$

$$\mu_{bi'}(y) = s[\mu_{B_i'}(y), \dots, \mu_{B_m'}(y)] \quad (10)$$

비퍼지화기에는 식 (11), 식 (12)와 같이 무게 중심법을 사용하였다.

$$y^* = \frac{\int_{V_a} y \mu_B(y) dy}{\int_{V_a} \mu_B(y) dy} \quad (11)$$

$$V_a = \{y \in V | \mu_B(y) \geq \alpha\} \quad (12)$$

표 2와 그림 12는 임시목표까지 주행을 위한 회전각도 출력에 대한 소속 함수와 제어 규칙을 나타내고 있다.

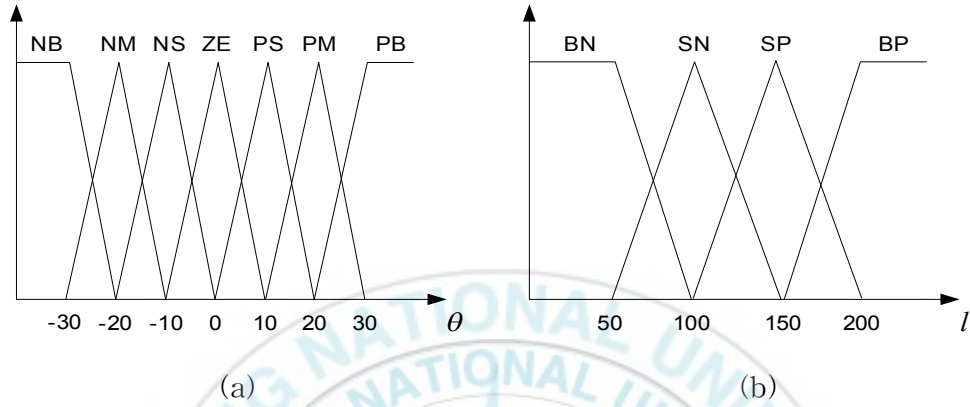


그림 12. 경로 설정 소속 함수
(a) 각도 소속 함수 (b) 거리 소속 함수

표 2. 회전각도 출력을 위한 경로 계획 제어 규칙

		Distance			
		BN	SN	SP	BP
A n g l e	NB	NB	NB	NB	NM
	NM	NB	NM	NM	NS
	NS	NM	NM	NS	NS
	ZE	ZE	ZE	ZE	ZE
	PS	PM	PM	PS	PS
	PM	PB	PM	PM	PS
	PB	PB	PB	PB	PM

표 3과 그림 13은 임시목표까지 빠른 주행을 위한 이동거리 출력에 대한 소속 함수와 제어 규칙을 나타내고 있으며, 임시목표와 로봇의 진행방향과의 차이는 절대값으로 하였다.

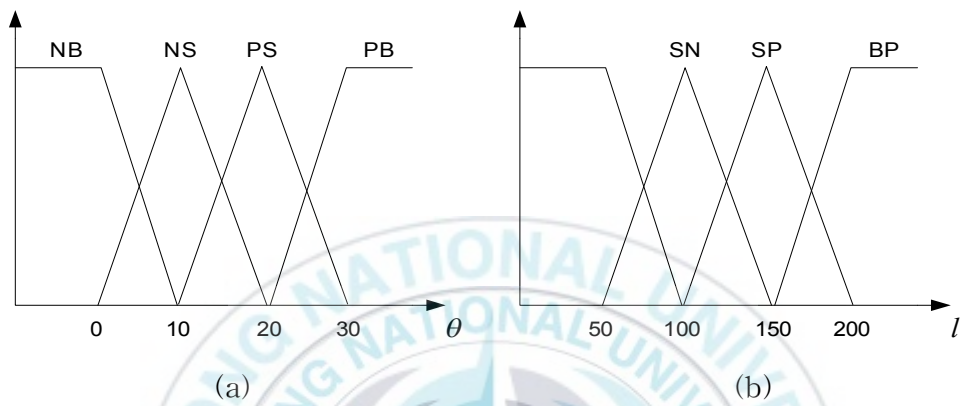


그림 13. 경로 설정 소속 함수

(a) 각도 소속 함수 (b) 거리 소속 함수

표 3. 이동거리 출력을 위한 경로 계획 제어 규칙

		Distance			
A n g l e		BN	SN	SP	BP
	NB	SN	SP	BP	BP
	NS	SN	SP	BP	BP
	PS	BN	SN	SN	SN
	PB	BN	SN	SN	BN

표 4와 그림 14는 장애물 회피를 위한 이동거리 출력에 대한 소속 함수와 제어 규칙을 나타내고 있다

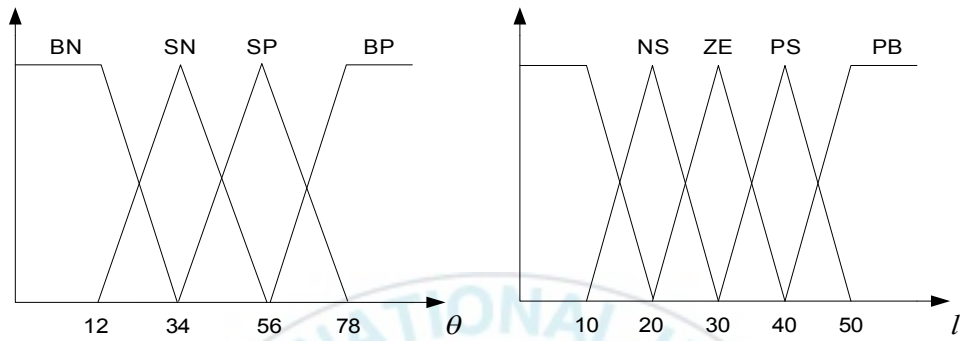


그림 14. 장애물 회피 소속 함수

(a) 각도 소속 함수 (b) 거리 소속 함수

표 4. 이동거리 출력을 위한 장애물 회피 제어 규칙

		Angle			
D i s t a n c e		BN	SN	SP	BP
	NB	NB	NB	NB	NS
	NS	NS	NS	ZE	ZE
	ZE	NS	ZE	ZE	PS
	PS	ZE	ZE	PS	PS
	PB	ZE	PS	PS	PB

표 5와 그림 15는 장애물 회피를 위한 회전각도 출력에 대한 소속 함수와 제어 규칙을 나타내고 있다.

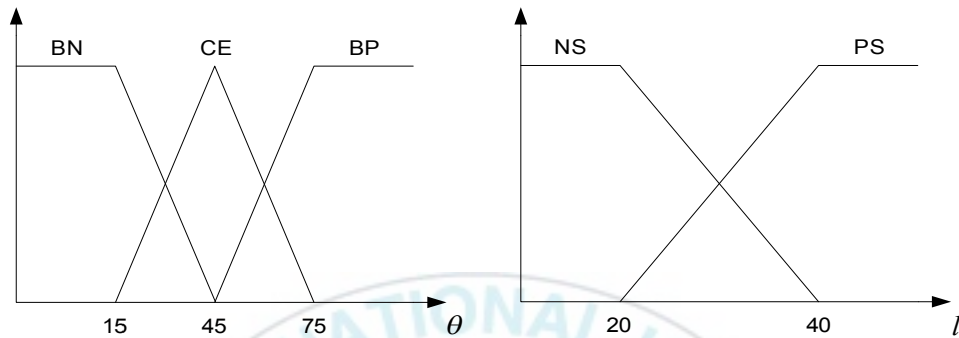


그림 15. 장애물 회피 소속 함수

(a) 각도 소속 함수 (b) 거리 소속 함수

표 5. 회전각도 출력을 위한 장애물 회피 제어 규칙

		Angle		
Distance		BN	CE	BP
	NS	CE	BN	BN
	PS	CE	CE	BN

그리고 현재 임시목표점이 카메라의 사각지대에 가까이 왔을 때 주행을 위한 제어권을 IR 센서로 넘겨주게 되어있다. 그리고 항상 장애물의 전방 꼭지점을 가리키게 되어있어 새로운 임시 목표로 전환이 이루어지지 않으면 현재 장애물의 외각만을 회전하는 현상이 발생하게 된다. 따라서 장애물 회피 동작이 이루어짐과 CCD 센서를 통한 새로운 임시목표의 검색이 이루어지도록 하였다.

3.2 위치 추정 알고리즘

3.2.1 노드 인식

구현된 로봇이 화재 진압용 로봇으로 행동하기 위해서는 실제로 화재 발생 지역을 인식하고 화재 진압을 위해 해당 지역으로 이동을 해야 한다. 그러기 위해 화재 발생 예상지역에 온도, 조도, 습도의 센서를 장착한 노드들을 위치시켰고 이동로봇과 Zigbee를 이용하여 센서 네트워크를 구성하였다. 각각의 노드들은 End Device로 동작을 하면서 센싱된 정보를 Coordinator로 동작하는 이동로봇으로 전송해준다. 로봇은 들어온 정보를 바탕으로 화재 발생 지역을 인식하고 해당 노드로 주행하게 된다.

로봇에 위치한 Coordinator와 각각의 End Device들 간의 Command와 Data등의 명령들과 Air중에 있는 Packet들을 Catch하여 지금 어떤 Packet이 송수신되고 있는지 Packet Sniffer를 통해서 확인할 수 있다. 그림 16은 이 Packet Sniffer를 이용하여 End Device에서 Coordinator로 전송하는 Data와 Coordinator에서 End Device로 전송하는 Data를 Capture하여 나타낸 것이다. End Device에서 Coordinator로 노드 B에서 화재가 발생하여 B의 아스키코드 42를 전송하고 있다. Address부분에서 0x0000은 Coordinator의 주소를, 0X796F~0XFFFF는 End Device 주소를 나타낸다.

원하는 노드로의 이동 방법은 노드의 좌표값, 로봇의 좌표값의 유무에 따라 달라진다. 구현된 시스템의 위치 추정을 위해 RSSI값을 이용한 삼각측량방법이 있으나 이는 정확성이 현저하게 떨어지므로 다음과 같은 2가지 방법으로 실험을 하였다.

Dest. Address	Source Address	APS Frame control field				APS Payload
0x0000	0x7970	Type	Del.mode	Ind.am	Sec Ack	21 AA 04 42
		Data	Indirect	1	0 1	42 42 42

Dest. Address	Source Address	APS Frame control field				APS Payload
0x796F	0x0000	Type	Del.mode	Ind.am	Sec Ack	21 BD 14 76 61 72 20 32 34 2C 20 32
		Data	Unicast	1	0 1	0D 76 61 72 20 32 35 2C 20 38 0D

그림 16. Packet Sniffer 실행모습

3.2.2 Motor Encoder를 이용한 위치 추정

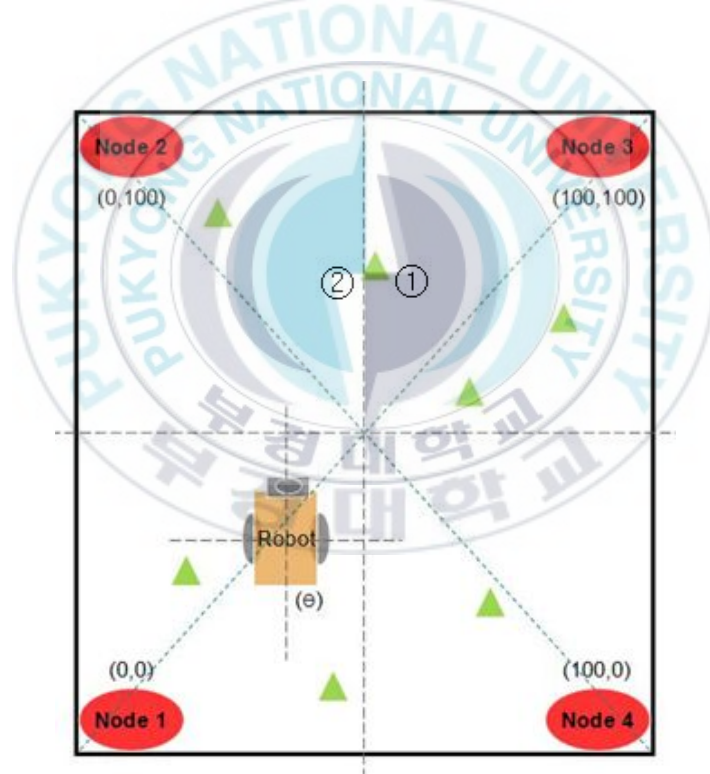


그림 17. Motor Encoder를 이용한 위치 추정

그림 17에서 삼각형은 장애물을 의미하고 원은 노드를 의미하며, 4개의 노드를 사용하였다. 노드를 각 사분면의 끝점인 45도, 135도, 225도, 315도

에 위치시키고, 로봇은 전방 90도를 향하도록 한다. 그런 후 로봇이 이동을 하면서 Motor Encoder를 통해 들어진 각도(θ)를 계속적으로 계산해낸다. 만약에 로봇이 움직이다가 현재 노드 3번을 향하고 있는데 노드 2번 지역에 화재가 발생했다고 가정하자. 그러면 θ 값과 노드 2번의 위치각도 정보인 135도를 로봇이 알고 있으므로 Temporary Target을 정할 때 기존의 방법에 원하는 노드로의 방향에 대한 가중치를 추가한다. 그렇게 되면 그림에서 ①이 임시목표점이 되는 것보다 ②가 임시목표점이 될 확률이 높게 되고 결국 로봇은 이와 같은 반복적인 작업을 통해 화재 발생 지역인 2번 노드로 찾아갈 수 있게 된다.

3.2.3 RSSI값을 이용한 위치 추정

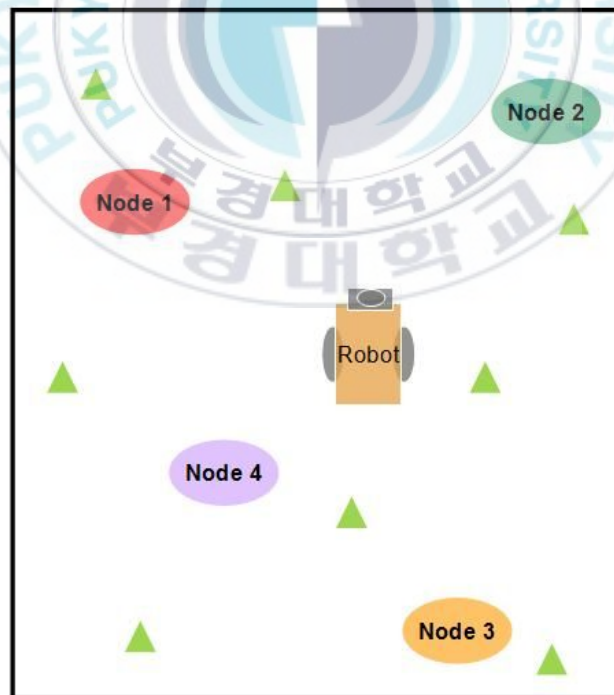


그림 18. RSSI값을 이용한 위치 추정

로봇이 이동하거나 정지할 때의 기준으로 LQI(Link Quality Indicator)를 사용한다. LQI는 RSSI(Received Signal Strength Indicator), 즉 수신신호 강도 값을 바탕으로 산출하는 값으로 신호의 강도와 질을 계산한 값이다. 본 시스템은 이러한 LQI값을 사용해서 거리측정을 시도하였다.

그림 18과 같이 노드와 로봇을 임의의 지점에 위치시킨다. 각각의 노드로부터 RSSI값을 로봇이 비교하면서 화재가 발생한 노드 쪽의 RSSI값이 점점 더 커지는 쪽으로 이동한다. 그리고 불안정한 삼각측량법이 아닌 각 노드들로부터 RSSI값을 받아서 그 값이 원하는 노드의 LQI값이 커지는 쪽으로 주행하도록 하였으며, 이 때 RSSI값에 LQI값이 급격하게 변화는 현상에 대처하기 위해 적절한 필터링을 통하여 확실한 값만을 받아들이도록 하였다. 만약에 노드 4번 지역에 화재가 발생했다고 가정하자. 그림에서는 이 거리가 가깝지만 실제로는 거리가 상당히 멀다고 하면 로봇은 해당 지역으로 이동을 완료했다고 판단해버릴 수도 있다. 따라서 원하는 노드의 Landmark를 확인할 때까지 이동을 계속적으로 하고 Landmark를 인식한 후에는 Landmark를 Target으로 설정해서 주행하도록 하였다.

IV. 실험 및 실험 결과

4.1 실험 환경

실험을 위해 사용된 시스템은 ARM9 계열의 삼성의 S3C2440 CPU가 탑재된 LN2440SBC 보드를 사용하여 Windows CE를 포팅하여 제안된 알고리즘을 구현하였다. 그리고 무선 센서 네트워크 구성을 위해 Zigbee Module인 한백전자의 Zigbex를 사용하였다. 경로 설정을 위한 이미지 처리를 위해 Logitech pro 4000p 웹캠을 사용했으며, 장애물 회피를 위해 10~80cm까지 거리측정이 가능한 PSD센서(GP2Y0A21YK)를 사용하였다. 그리고 SST42D2120 스텝모터를 사용했으며, 모터 제어와 통신을 위해 ATmega128 프로세서를 사용하였다.

그림 19는 실험에 사용한 인터페이스 화면을 보이고 있다.

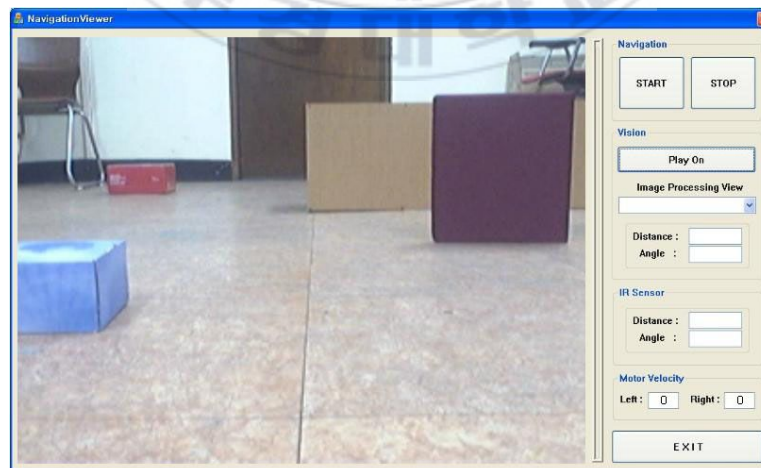


그림 19. 실험에 사용한 인터페이스

4.2 주행 및 장애물 회피 결과

그림 20은 본 논문에서 제안한 영상 처리 순서에 의해 얻어진 각각의 영상을 보였으며, 마지막 그림에서 흰 부분이 장애물로 검출된 부분이다.



그림 20. 영상 처리 과정

그림 21은 영상정보에 의해 임시목표를 잡고 가속으로 목표점까지 이동한 후 장애물을 회피하는 것을 보여주고 있다. 그림 22는 장애물 회피 시에 새로운 임시목표를 잡고 경로설정을 하고 있다. 이는 결과에서 나타난 것과 같이 비전에 의한 주행과 IR 센서를 통한 주행이 서로 잘 융합되었음을 알 수 있고, 구현된 제어기로 인한 빠른 주행과 경로 계획, 그리고 장애물 회피 단계로 자연스럽게 이어지는 것을 확인할 수 있다. 그림 23은 좀 더 복잡한 환경에서의 자율주행을 보여주고 있으며, 그림에서 보는 바와 같이 장애물을 피해 잘 주행하고 있음을 보여준다.



그림 21. 주행 및 장애물 회피 실험 1



그림 22. 주행 및 장애물 회피 실험 2

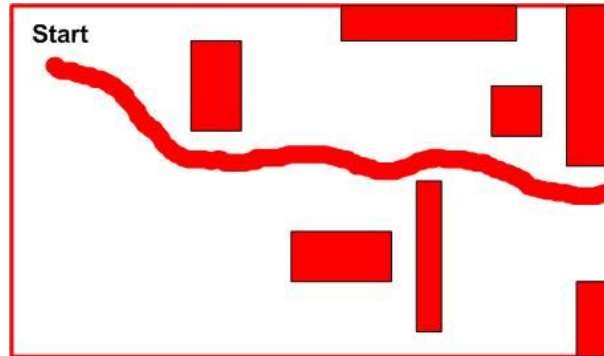


그림 23. 주행 및 장애물 회피 실험 3

4.3 노드 인식 및 위치 추정 결과

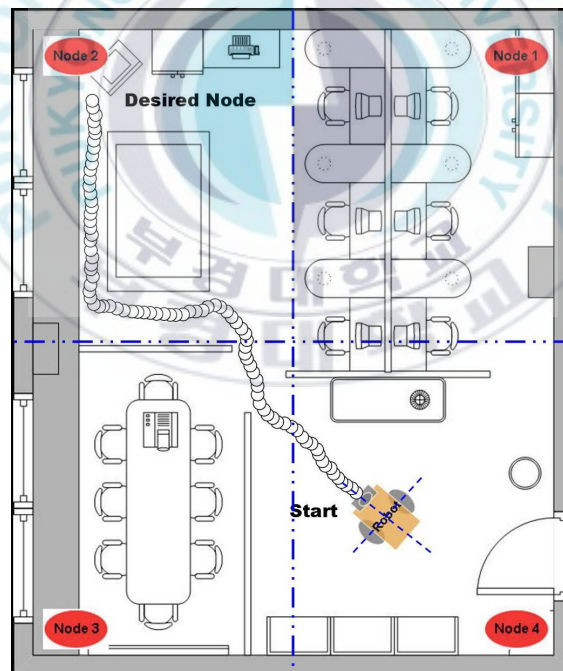


그림 24. Motor Encoder를 이용한 주행 결과

실제 생활공간에서 화재 발생 지역을 인식하고 해당 지역으로 이동하는

모습을 보이고 있는데, 그림 24는 제안된 노드 인식 및 주행 방법 중 엔코더 값을 이용한 결과이다. 그림에 나타난 것과 같이 엔코더를 이용한 주행에서는 구현된 제어기로 인한 빠른 주행과 경로 계획, 위치추정 그리고 장애물 회피 단계로 자연스럽게 이어지는 것을 확인할 수 있다.

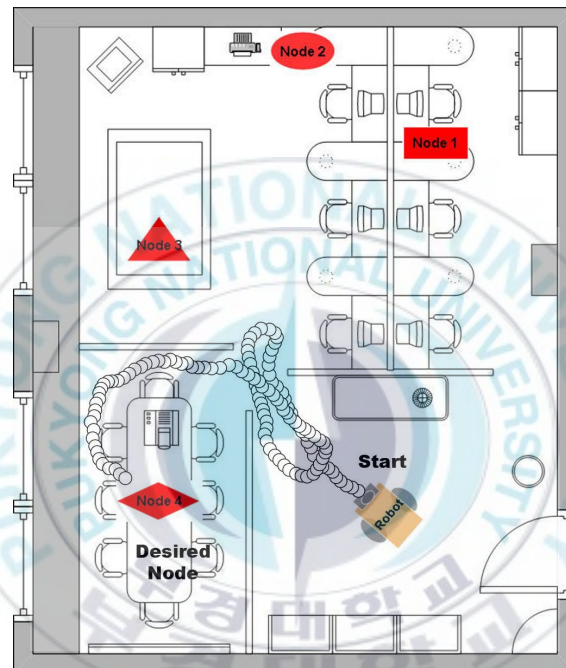


그림 25. RSSI값을 이용한 주행 결과

그림 25는 RSSI 값을 통한 주행 결과이다. RSSI 값의 흔들림 정도가 너무 커서 원하는 목표지점으로 찾아가는데 보다 많은 시간이 걸리는 것을 확인할 수 있다. 또한 본 시스템은 이러한 LQI값을 사용해서 거리측정을 시도하였는데, 이 값이 RSSI 값이 변함에 따라 민감하게 변하지 않는 문제가 발생하였다. 예를 들면 RSSI가 -100에서 -50까지는 LQI = 80으로 변환해서 상위Layer(MAC)로 올려주고 -49부터 -20까지는 LQI = 120으로

변환해서 상위 Layer로 올려주는 것이었다. 결국 이 값이 튀는 현상의 원인이었다. 즉, 거리를 조금씩 움직이면 LQI값이 움직인 거리만큼 변환되어야 하는데 그렇지 못하다가 일정 거리 이상 멀어지면 급격하게 상승하거나 급격하게 감소하는 것이 문제였다. 이 문제를 해결하기 위해 송신전력을 바인딩을 할 수 있는 최소한의 전력으로 낮춰서 전송하고 중간 값은 모두 Discard하고 목적 노드와 로봇이 가까이 왔을 때의 일정 값만을 받아 로봇을 정지하게 해봤지만 이 방법도 만족할 만한 결과를 만들어 내지 못했다.



V. 결 론

본 논문에서는 화재 진압 로봇을 위한 이동 로봇의 자율 주행 방법을 제안하였다. 실험 결과에 나타난 것과 같이 장애물 인식 및 임시목표 설정이 잘 이루어지고 있음을 확인할 수 있었고, Vision과 IR 센서를 통한 퍼지 제어기 역시 잘 동작하는 것을 확인하였다. 그리고 실제 환경에서 화재 발생 예상 지역에서 노드를 설치하고 그 노드들과 로봇과의 무선 센서 네트워크를 통한 노드 인식 과정 역시 만족할 만한 결과를 얻을 수 있었다. 위치 추정 방법으로 제안했던 Motor Encoder를 이용한 주행 방법은 실험 결과에 나타난 바와 같이 잘 동작하는 것을 알 수 있었으나, RSSI값을 이용한 주행은 아직 불안정한 요소가 많아서 만족스런 결과를 얻지는 못하였다.

따라서 보다 정확하고 빠른 위치 추정 알고리즘이 개발되어야 할 것이고, Server 구현을 통해 Web Monitoring 및 Web Control 기능을 추가하여 보다 완성도 높은 화재 진압 로봇이 되어야 할 것이다. 또한 추가적으로 빠르고 정확한 주행 및 사용자 맵 모니터링을 위해 Map Building 과정도 필요하겠다.

VI. 참고 문헌

- [1] Xu, W.L., Tso, S.K., "Sensor-based fuzzy reactive navigation of a mobile robot through local target switching" IEEE Transaction on systems, man, cybernetics vol 9, no 3, pp.451-459, 1999
- [2] D. R Ramirez, D. Limon,, J Gomez-Ortera and E. F Camacho, "Nonlinear MPBC for Mobile Robot Navigation Using Genetic Algorithm", Proc. of the 1999 IEEE International Conference on Robotics and Automation Detroit, Michigan, pp. 2458-2464, 1999
- [3] Dilip Kumar Pratihari, Kalyanmoy Deb, and Amitabha Ghosh, "Fuzzy-Genetic Algorithms and Mobile Robot Navigation Among Static Obstacle", IEEE Transaction on systems, man, cybernetics, pp. 327-333, 1999
- [4] Akihisa Ohya, Akiko Kosaka and Avinash Kak, "Vision-Based Navigation by a Mobile Robot With Obstacle Avoidance Using Single-Camera Vision and Ultrasonic Sensing", IEEE Transaction on Robotics and Automation, vol. 14 no.6, pp. 969-978, 1998
- [5] Rafael C. Gonzalez, Richard E. Woods, *Digital Image Processing*, Addison Wesley, 1993

- [6] 김유찬, 유영재, 장영학, 송정곤, 이주상 “지능형 이동로봇을 위한 PSD 센서기반 거리계측 시스템의 개발” 한국 퍼지 및 지능시스템학회 2005년도 춘계학술대회 학술발표 논문집, 15권 1호, pp. 225-228, 2005
- [7] Jin-Woo Park, Jae-Han Park, Jang-Myoung Lee, "Intelligent Control of a Mobile Robot using Fuzzy Logic", Proceeding of ITC-CSCC '99, pp. 585-588, 1999
- [8] IEEE, "Wireless Medium Access Control(MAC) and Physical Layer (PHY) specifications for Low Rate Wireless Personal Area Networks", IEEE std, 802.15.4, 2003
- [9] Luo, R.c.; Ogst Chen; Pan, S.H, "Mobile User Localization in Wireless Sensor Network Using Grey Prediction Method" Industrial Electronics Society, 2005, IECON 2005, 32nd Annual Conference of IEEE 6-10 Nov. 2005