

논문 2008-45SC-6-20

# 단일 영상과 거리센서를 이용한 SLAM 시스템 구현

(Implementation of the SLAM System Using a Single Vision and Distance Sensors)

유 성 구\*, 정 길 도\*\*

(Sung Goo Yoo and Kil To Chong)

## 요 약

SLAM(Simultaneous Localization and Mapping)은 무인 로봇 자동 항법시스템의 중요 기술로 센서 데이터로부터 로봇의 위치를 결정하고 기하학적 맵을 구성하는 것이다. 기존 방법으로는 초음파, 레이저 등의 거리 측정 센서를 이용해 로봇의 전역 위치를 찾는 방법과 스테레오 비전을 통한 방법이 개발되었다. 거리 측정 센서만으로 구성한 SLAM 시스템은 계산량이 간소하고 비용이 적게 들지만 센서의 오차나 비선형에 의해 정밀도가 조금 떨어진다. 이에 반해 스테레오 비전 시스템은 3차원 공간영역을 정확히 측정할 수 있지만 계산량이 많아 고사양의 시스템을 요구하고 스테레오 시스템 또한 고가이다. 따라서 본 논문에서는 단일 카메라 영상과 PSD(position sensitive device) 센서를 사용하여 SLAM을 구현하였다. 전방향의 PSD 센서로부터 일정 거리의 장애물을 감지하고 전면 카메라의 영상처리를 통해 장애물의 크기 및 특징을 감지한다. 위의 데이터를 통해 확률분포 SLAM을 구성하였고 실제 구현을 통해 성능검증을 하였다.

## Abstract

SLAM(Simultaneous Localization and Mapping) system is to find a global position and build a map with sensing data when an unmanned robot navigates an unknown environment. Two kinds of system were developed. One is used distance measurement sensors such as an ultra sonic and a laser sensor. The other is used stereo vision system. The distance measurement SLAM with sensors has low computing time and low cost, but precision of system can be somewhat worse by measurement error or non-linearity of the sensor. In contrast, stereo vision system can accurately measure the 3D space area, but it needs high-end system for complex calculation and it is an expensive tool. In this paper, we implement the SLAM system using a single camera image and a PSD sensors. It detects obstacles from the front PSD sensor and then perceive size and feature of the obstacles by image processing. The probability SLAM was implemented using the data of sensor and image and we verify the performance of the system by real experiment.

**Keywords :** localization, mapping, single vision, probability

## I. 서 론

로봇 기술이 발달함에 따라 단순 반복 기능의 산업 로봇에서, 인간의 일상생활에 도움을 주는 서비스 로봇,

엔터테인먼트 로봇으로 발전하고 있다. 이러한 로봇들은 필수적으로 자율이동(Autonomous Navigation)기능을 탑재하고 있어야 한다. 자율이동이란, 로봇 스스로 현재위치를 파악하고 목적지까지 경로추적을 통해 이동하는 기능을 말한다<sup>[1]</sup>.

로봇의 위치파악 방법에는 크게 실외와 실내로 나눌 수 있는데, 실외의 경우 GPS, DGPS와 같은 위성항법 장치를 통해 위치를 파악하며, 실내의 경우 전역초음파 시스템이나 SLAM(Simultaneous Localization and Mapping)을 활용한다. 본 논문에서는 SLAM을 활용하여 자율이동로봇을 구성하였다. SLAM은 로봇이 주행

\* 정희원, 전북대학교 제어계측공학과  
(Control and Instrumentation Department, Chonbuk National University)

\*\* 정희원, 전북대학교 전자정보  
(Electronics and Information Department, Chonbuk National University)

※ 이 논문은 전라북도 지원으로 연구되었음.  
접수일자: 2008년8월26일, 수정완료일: 2008년10월22일

하는 동안 얻은 관찰과 제어신호를 가지고 주행한 환경을 지도화하며 동시에 자신의 절대 위치를 추정하는 것이다<sup>[2~3]</sup>. SLAM을 할 때 주변 환경에 대한 정확한 정보는 매우 중요하다. 환경관측의 정확성, 데이터 처리 과정 및 시간 등이 주요 요소라고 할 수 있다. 기존의 많은 SLAM 알고리즘은 관측시스템으로 센서를 사용하였다. Crowley<sup>[4]</sup>는 칼만 필터형태 알고리즘을 사용하여 초음파 기반 자율이동 로봇을 제안하였고, Nuchter<sup>[5]</sup>는 LRF센서를 적용한 다차원 맵핑을 제안하였다. 이러한 센서기반 방법은 센서잡음에 민감하고 맵 구성시 점이나 선과 같은 간단한 기하학적인 원소로 표현하기 때문에 한 번의 관찰로는 전체 맵의 대응 관계를 찾는 것은 어렵다. 따라서 비전 기반 센서를 이용해 대응 관계를 찾는 연구가 수행되었다. Stephen<sup>[6]</sup>은 영상에서 랜드마크를 찾아 맵을 생성하고 대응관계를 찾는 방법을 제안하였고, Davison<sup>[7]</sup>은 단일 카메라를 낮은 속도로 이동하여 특징점을 찾는 방법을 제안하였다. 하지만 오직 비전 기반의 추적 형태로 대응관계를 구해 지도를 생성하면 지도를 밀집하게 표현하여 가려짐 같이 추적을 잃은 상황에서 대응관계를 찾을 수 없다. 이를 해결하기 위해 Lowe<sup>[8]</sup>은 연속영상에서 인접한 프레임 상의 특징량을 추출하는 방식으로 접근하였다. 그리고 Thrun<sup>[9]</sup>은 스테레오 영상으로부터 3D 특징량을 추적하여 대응관계를 찾는 방법을 제안하였다. 하지만 위의 방법은 데이터 처리량이 많기 때문에 로봇의 이동 속도를 보장하기 어렵고 고사양의 제품사양을 요구하기 때문에 경제적인 측면에서도 불리하다<sup>[10]</sup>.

따라서 본 논문에서는 정밀도가 높은 PSD(Position Sensitive Device) 센서와 단일 카메라를 사용하여 기존의 단점을 보완한 시스템을 구현하고 성능을 검증하였다. 로봇 전방에 45도 간격으로 5개의 PSD 센서를 장착하였으며 30fps 성능의 1394카메라를 장착하였다. 센서로부터 일정거리 안의 장애물이 발견되면 카메라로부터 전방 영상을 획득하고 특징점 처리를 통해 장애물의 크기와 위치를 확인한다. 센서정보와 영상정보로부터 확률분포 SLAM 기법을 통해 주변 환경지도를 생성한다. 이를 위해 실제 이동 로봇을 제작하였으며 밀폐된 공간에서 장애물을 설치하여 로봇을 자율 이동시켜 성능을 검증하였다.

본 논문의 구성은 다음과 같다. II장에서는 실험로봇과 전체적인 시스템 구성을 언급하였고, III장에서는 영상 정보로부터 환경정보를 추출하는 방법에 대해 기술하였다. IV장에서는 환경지도를 생성하는 확률분포

SLAM에 대해 설명하였으며 V장에서는 실험 결과와 성능비교에 대해 기술하였다. 마지막으로, VI장에서는 결론을 내리고 추후 해결 과제에 대해 기술하였다.

## II. 로봇 모델 및 시스템 구성

### 2.1. 모바일 로봇 시스템

본 연구를 위해 제작한 로봇은 그림 1과 같다.

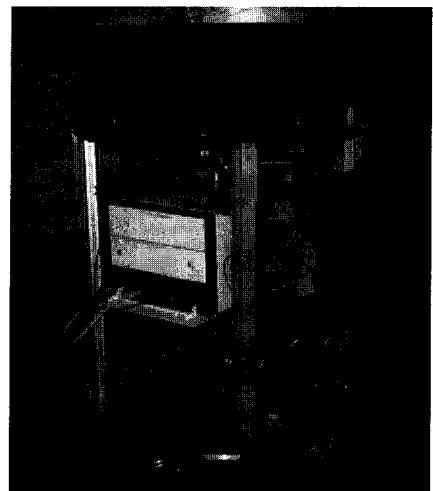


그림 1. SLAM 이동 로봇  
Fig. 1. Mobile robot for SLAM.

5개의 PSD 센서를 전면에 장착하였으며, 두 개의 BLDC모터를 장착하여 로봇의 이동을 제어하며 중앙에 제어기를 설치하였다. 각 바퀴에는 엔코더를 장착하여 SLAM의 제어기 입력으로 사용하였다. 앞과 뒷면의 바닥에는 옴니디렉션바퀴를 장착하여 모바일 로봇의 자율 이동이 가능하도록 하였다. 로봇의 모델링 방정식은 다음과 같다.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \quad (1)$$

여기서  $x, y$ 는 로봇의 중점의 위치,  $\theta$ 는 축을 기준으로 한 목표지점간의 각도를 나타낸다. 선형 속도  $v$ 와 각속도  $w$ 는 다음과 같다.

$$v = \frac{v_l + v_r}{2}, \quad w = \frac{v_l - v_r}{D} \quad (2)$$

여기서  $v_l$ 과  $v_r$ 은 각각 좌측 바퀴와 우측 바퀴의 속도를 나타내며,  $D$ 는 두 바퀴간의 거리를 나타낸다.

## 2.2. SLAM 시스템 구성

SLAM을 구현하기 위한 전체 시스템 구성은 그림 2와 같다. PSD 센서로부터 일정간격 안에 장애물이 발견될 시 카메라로부터 영상을 받고 영상처리를 통해 장애물의 특징을 추출해 낸다. 센서데이터와 영상데이터를 이용해 맵을 생성한 후 이전 위치와 현재의 엔코더데이터를 이용하여 위치 보정을 실행한다. 로봇의 이동에 따라 위의 과정을 반복하여 전체 맵을 구성한다.

제어 시스템은 LABVIEW 7.0을 이용하여 구성하였다. DAQ보드를 이용하여 센서정보와 엔코더 정보를 획득하며 NI Vision Builder를 사용하여 영상처리를 구현하였다. 제어프로그램의 전체적인 구성은 그림 3과 같다.

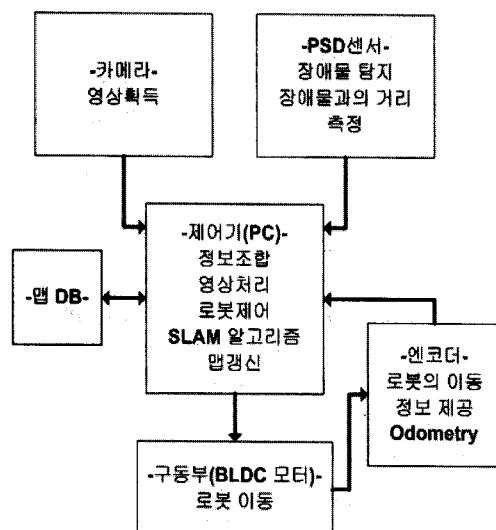


그림 2. 시스템 구성도

Fig. 2. Configuration of the System.

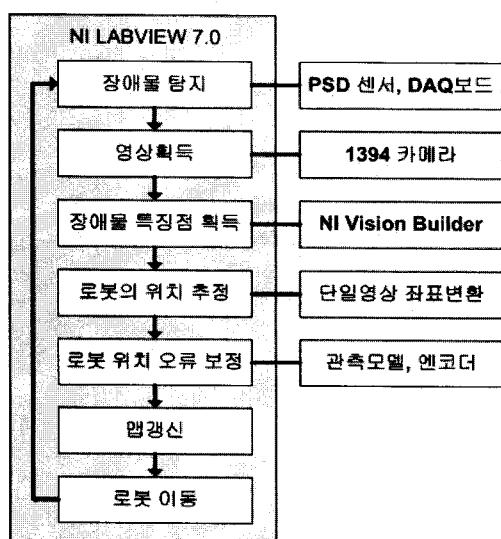


그림 3. 제어프로그램 순서 및 구성

Fig. 3. Flow chart and contents of the control program.

## III. 영상시스템과 위치 관계

본 논문에서는 센서정보를 이용하여 영상에서 장애물을 찾고 특정점 처리를 통해 장애물의 가로 크기만을 측정한다. 이는 맵의 구성을 2차원으로 하기 때문에 복잡하게 모든 영상의 윤곽선 검출을 시행하지 않고 영상에서 장애물을 탐지한 부분에서만 검출을 시행한다. 이는 영상처리 시간을 줄일 수 있고 시스템의 자원을 효율적으로 사용할 수 있다.

### 3.1. 이동로봇 좌표변환

로봇이 초기위치에서 이동시 엔코더를 사용하여 로봇의 좌표를 구할 수 있다. 하지만 바퀴의 슬립이나 충돌로 인해 좌표오류가 발생한다. 이를 보정하기 위해 영상으로부터 장애물과의 거리를 측정하여 엔코더 정보를 보완한다. 일반적으로 물체와의 거리를 계산하는 데는 2개의 카메라를 사용해 각각의 카메라에서 얻어진 영상을 비교하여 거리를 계산한다. 이러한 거리 측정 방식을 스테레오 비전방식이라고 하는데, 스테레오 비전 방식은 연산시간이 오래 걸려 실시간으로 장애물을 인식하고 회피하는데 불리하다. 따라서 본 논문에서는 단일 카메라를 사용하여 연산시간을 줄이고 물체와의 거리를 계산하였다. 단일 카메라로부터 3차원 공간상의 좌표를 계산하기 위해 모든 물체는 바닥에 접하고 있다고 가정하여 2차원 평면공간으로 해석하였다<sup>[11]</sup>.

그림 4와 5는 좌표변환 방법을 도시한 내용으로, 바닥에서 카메라까지의 높이  $h$ , 로봇 앞부분에 존재하는 사각 지역의 거리  $y_b$ , 카메라에서 보이는  $y$ 축 방향의 거리  $y_l$ ,  $x$ 축 방향의 거리  $x_l$ , 위의 네 가지 요소를 이용하여  $\alpha, \beta, \theta$  각을 구할 수 있다.

$$\alpha = \tan^{-1} \left( \frac{h}{y_b} \right) \quad (3)$$

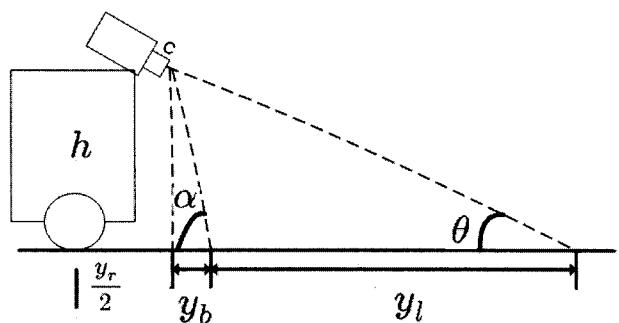


그림 4. 좌표변환 측면

Fig. 4. Side view of the transfer coordination.

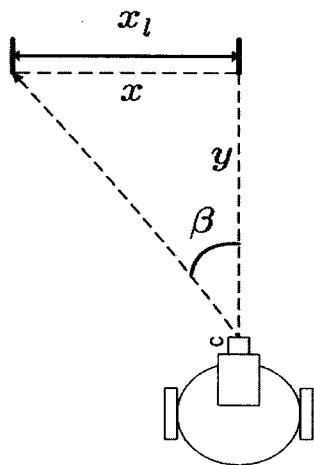


그림 5. 좌표변환 윗면

Fig. 5. Top view of the transfer coordination.

$$\beta = \tan^{-1} \left( \frac{h}{y_b + y_l} \right) \quad (4)$$

$$\theta = \tan^{-1} \left( \frac{y_b + y_l}{h} \right) \quad (5)$$

위의 각을 통해 영상기반의 좌표를 이동로봇 기반 좌표로 변환할 수 있다.

$$y = \frac{h}{\tan \left( \theta + \frac{y_p(\alpha - \theta)}{y_s} \right)} + \frac{y_r}{2} + y_b \quad (6)$$

여기서,  $y_p$ 는 영상에서 y의 픽셀 개수이며,  $y_s$ 는 y축 전체 픽셀 개수이다.  $y_r$ 은 로봇의 y축 길이이다. 또한 x좌표는 식(7)을 이용하여 구할 수 있다.

$$x = \tan \left( \frac{\beta(1 - 2x_p)}{x_s} \right) \times y \quad (7)$$

여기서,  $x_p$ 는 x의 픽셀 개수이며,  $x_s$ 는 x축 전체 픽셀 개수이다. 식(6)과 (7)을 이용하여 로봇의 좌표  $(x, y, \theta)$ 를 구할 수 있다.

#### IV. 확률분포 SLAM

SLAM 알고리즘은 주로 확률적으로 계산된다. 즉 현재의 정확한 위치를 추정하는 것이 라기 보다는 어느 영역에 어떤 확률 분포를 가지는지를 알아내는 것이다. 확률 분포 SLAM 방법론이 주로 사용되는 이유는 오차 누적을 방지할 수 있기 때문이다. 로봇의 위치를 엔코더

와 같이 하나의 정보만으로 결정하였을 경우 오차가 발생하게 된다면 오차 보정을 해줄 수 없으므로 오차가 누적되게 된다. 하지만 실제 위치를 항상 정확히 알아낼 수 없기 때문에 다른 위치에서도 확률분포값을 가지고 로봇이 진행하면서 여러 위치에 대해서 다시 확률 값을 계산함으로써 이전에 로봇의 위치가 잘못 계산되었을 경우에도 현재 위치를 보다 정확히 알아낼 수 있다. 즉 현재의 로봇 및 장애물의 정확한 위치를 계산하는 것이 아닌 위치에 대한 확률 값을 가지고 있기 때문에 다음 단계에서의 위치를 다시 알아낼 수 있다.

##### 4.1. SLAM의 확률 모델

SLAM의 확률 모델은 식(8)과 같다<sup>[12]</sup>.

$$P(s_t, m | Z^t, U^{t-1}) \quad (8)$$

$$s_t = [x_t, y_t, \theta_t]^T,$$

$$m = \{x_{m1}, y_{m1}, x_{m2}, y_{m2}, \dots, x_{mn}, y_{mn}\}$$

여기서  $s_t$ 는 현재 로봇의 위치와 진행방향을 뜻하고,  $m$ 은 현재까지 저장된 특징점들의 위치이다.  $Z^t$ 는 현재까지의 센서입력,  $U^{t-1}$ 은 엔코더로부터 받은 값, 즉 이전의 로봇 위치로부터 현재 위치에 대한 상대적인 움직임에 대한 입력값을 나타낸다.

식(8)을 다시 표현한다면,

$$P(s_t, m | Z^t, U^{t-1}) = \int P(s_t | s_{t-1}, u_t) \times P(s_{t-1}, m | Z^{t-1}, U^{t-2}) dx_{t-1} \quad (9)$$

여기서,

$$m = \{m_k : k = 1, 2, \dots, N\}$$

$$Z^t = \{z_k : k = 1, 2, \dots, L\}$$

$$U^{t-1} = \{u_1, u_2, \dots, u_{t-1}\}$$

이다.

확률적인 SLAM은 식(9)로 표현되는 확률의 분포를 구하는 과정이다.

확률 분포를 구하기 위해서 무한대의 상태 변수에 해당하는 확률값을 구해야 하는데, 이는 실제 시스템에서 구현하는 것은 불가능 하다. 따라서 위의 확률식을 가우시안분포로 가정함으로써 만약 특징점의 수가  $N$ 일 경우  $2N+3$ 의 평균 벡터와  $(2N+3) \times (2N+3)$ 의 공분산 행렬(covariance matrix)로부터 확률 분포를 모두 설명할 수 있다<sup>[13~14]</sup>.

## 4.2 움직임 모델(Motion Model)

로봇의 움직임을 제어 신호에 따라 확률적으로 설명하는 모델로서, 이전에 추정한 로봇의 위치로부터 현재의 위치를 추정한다. 그리고 추정된 위치에서 바라본 관측 특징량과 지도 특징량의 대응 관계를 구한다. 여기서 관측 특징량은 3장에서 설명한 좌표값을 뜻한다.

일반적임 움직임 모델은 다음과 같다.

$$s_t = f(s_{t-1}, u_{t-1}, w_{t-1}) \quad (10)$$

여기서,  $s_{t-1}$ 은 이전 로봇 및 특징점의 위치에 대한 상태변수이고  $u_{t-1}$ 은 주행계로부터 얻은 입력이다. 그리고  $w_{t-1}$ 은 가우시안 노이즈이다. 전진(dist)과 회전(ang) 제어신호에 따른 움직임 모델을 표현하면 다음과 같다.

$$\begin{aligned} x(t+dt) &= x(t) + (u_{dist}(t) + w_{dist}(t)) \\ &\quad \times \cos(\theta(t) + (u_{ang}(t) + w_{ang}(t))/2) \\ y(t+dt) &= y(t) + (u_{dist}(t) + w_{dist}(t)) \\ &\quad \times \sin(\theta(t) + (u_{ang}(t) + w_{ang}(t))/2) \\ \theta(t+dt) &= \theta(t) + u_{ang}(t) + w_{ang}(t) \\ u_t &= [u_{dist}(t) \quad u_{ang}(t)]^T, \\ s_t &= [x_v(t) \quad y_v(t) \quad \theta_v(t)]^T \end{aligned} \quad (11)$$

## 4.3 관측 모델(Measurement Model)

관측 모델은 현재 로봇의 위치에서 기준의 만들어진 지도의 특징점들이 어떻게 관측될 것이라는 것을 예측하고 그 예측된 특징점과 현재의 영상으로부터 구한 특징점들 사이의 유사도를 표현하는 것을 의미한다. 로봇의 자세와 지도 특징량의 위치를 알고 있다면 예측한 지도 특징량과 관측 특징량을 한 좌표계로 옮긴 후 대응 관계를 구할 수 있다.  $x_L, y_L$ 은 지도 특징량의 위치를 나타내고,  $x_v, y_v$ 는 로봇의 위치,  $\theta$ 는 로봇의 방향을 나타낸다. 여기서 관측 모델은 다음과 같다.

$$\begin{bmatrix} x'_m \\ y'_m \end{bmatrix} = \begin{bmatrix} \sin\theta & \cos\theta \\ \cos\theta & -\sin\theta \end{bmatrix} \begin{bmatrix} x_L - x_v \\ y_L - y_v \end{bmatrix} \quad (12)$$

$$\theta'_m = \tan^{-1}\left(\frac{y'_m}{x'_m}\right)$$

현재 위치에서 획득한 영상으로부터 계산한 로봇의 위치와 관측모델을 통해 예측한 관측 위치를 비교하여 위치오차를 보정하고 시스템의 성능을 판별한다.

## V. 실험 및 결과

### 5.1. 시스템 사양 및 실험 환경

본 연구를 위해 제작한 로봇은 2장에서 언급하였듯이 2-wheel 모바일 로봇이며, 1m~5m 와 20cm~150cm의 거리 측정이 가능한 PSD 센서를 조합하여 사용하였다. 카메라는 ptgrey사의 Dragonfly<sup>®</sup>2로 1024×768의 해상도와 30FPS(Frame per Second)의 성능을 가지는 1394전용 카메라이다. 제어기 프로그램은 NI사의 LABVIEW 7.0을 통해 구현하였고, 영상획득 및 처리는 NI사의 Vision Builder 3.0을 사용하여 구현하였다. SLAM 구현을 위한 실험환경 구축은 그림 6과 같다. 크기는 20.5m<sup>2</sup>이며 4개의 장애물이 그림과 같이 놓여 있다.

### 5.2. SLAM 주행 결과

그림 6에서 보인 환경에서 자율주행 시험을 실시하였다. 로봇의 초기위치와 이동 동선은 그림 6에 나타내었다. 그림 7은 센서로부터 로봇을 회전시키면서 장애물을 감지하였을 경우 Vision Builder를 사용하여 장애물의 크기와 특징을 추정하는 과정을 보여준다. 이를 통해 장애물의 크기정보와 거리정보를 지도정보에 간신히 한다.

그림 8은 로봇을 이동 동선에 따라 한 바퀴를 이동한 후의 실제 이동선과 지도를 보여준다. 검은색 원으로 표시된 라인은 로봇의 이동모습을 보여주며 사각형으로 표현된 것은 장애물의 정보를 표시한다. 지도의 장애물

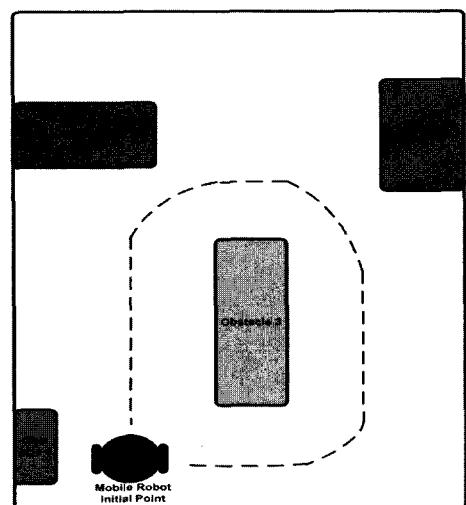


그림 6. 실험 환경 및 로봇 이동 동선

Fig. 6. Experiment environment and a moving line for robot.

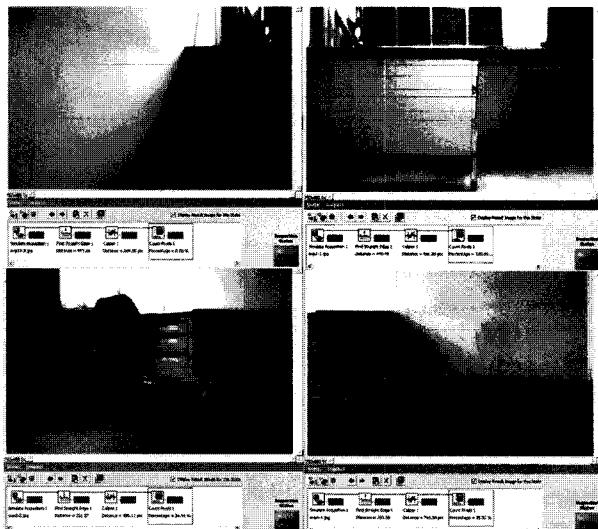


그림 7. Vision Builder를 이용한 장애물의 크기 측정

Fig. 7. Measurement of the obstacle size using Vision Builder.

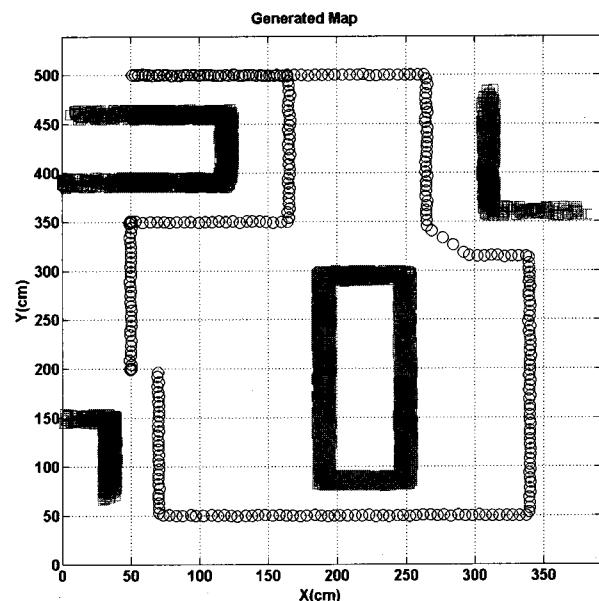


그림 9. 로봇 이동에 따라 생성된 전체 지도

Fig. 9. Total map after moving of the robot.

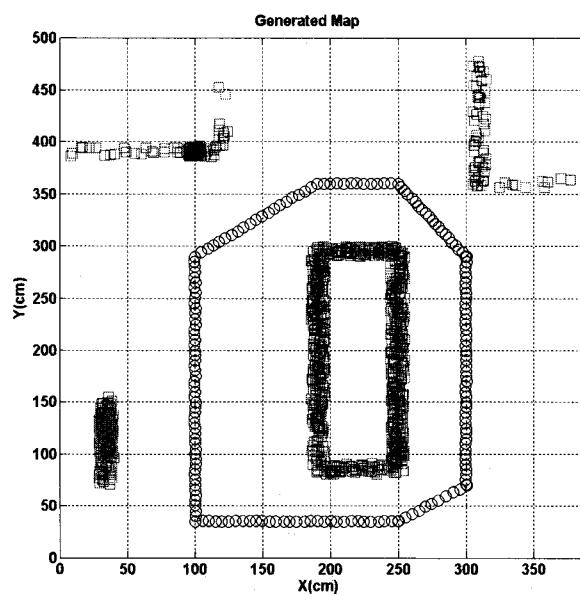


그림 8. 로봇 이동에 따라 생성된 지도

Fig. 8. Generated map after moving of the robot.

위치 오차평균은 약 10cm 정도로 로봇이 이동 횟수가 많아질수록 오차는 줄어든다. 지도에서 진하게 표시된 장애물은 확률분포가 높은 영역을 표시한다. 만약 이동 횟수를 증가한다면 높은 확률분포영역의 연결을 통해 정확한 지도를 생성할 수 있다. 하지만 좀 더 영역이 크고 복잡한 환경에서 SLAM을 실행할 경우 지도정보가 커짐에 따라 시스템의 처리시간이 길어질 수 있는 단점이 발생할 수도 있다. 따라서 일정 시간이 지난 후 일정 확률분포이상만을 가지는 특징점만을 추출하여 이를 통해 최종 지도를 생성해야 한다.

그림 9는 (50, 200)의 출발점을 시작으로 (70,200)까지

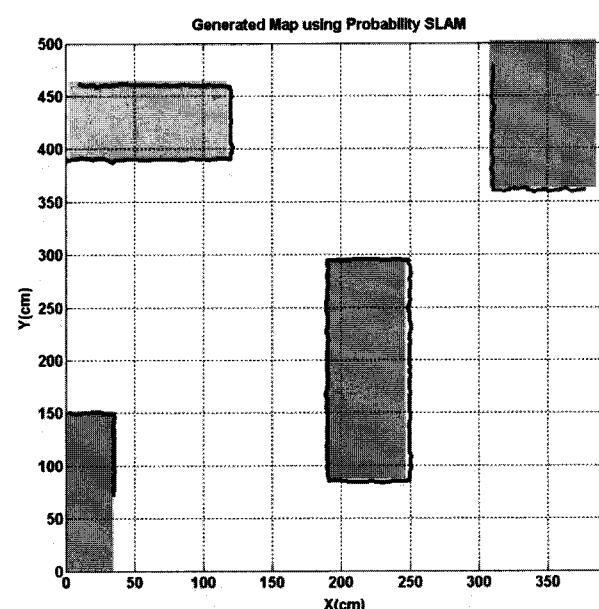


그림 10. 확률 SLAM을 적용한 전체 지도

Fig. 10. Global map using probability SLAM.

로봇이 벽면과 장애물을 탐지하면서 이동하여 최종적으로 생성해낸 지도이다. 평균오차는 약 7cm이며 카메라의 사각지역과 센서사각지역으로 인해 전체 장애물의 모습을 포착을 못할 경우가 발생한다. 이를 위해 일정 확률분포 이상의 포인트만을 연결하면 최종적인 지도를 획득할 수 있다.

그림 10은 일정 확률분포를 가지는 점들의 조합으로 최종적으로 지도를 완성한 것이다. 그림 6에 보인 실험 환경과 유사한 최종 지도를 완성하였다. 하지만 좌측하

단과 우측 하단의 장애물은 카메라와 센서의 사각으로 인해 실제보다 크게 측정된 것을 볼 수 있다.

### 5.3 실험 결과 비교

본 연구의 성능을 검증하기 위하여 타 시스템과의 성능을 검증하였다. 표 1에 각 시스템의 정확도와 처리속도를 나타내었다. 또한 시스템 구현시 필요한 비용적인 면을 보였다.

PSD와 초음파센서 시스템 실험은 직접 진행하였으며, 스테레오 비전<sup>[9]</sup>과 레이저센서<sup>[5]</sup>는 비교논문을 참조하였다. PSD는 빠른 처리시간과 비용적인 측면에서 장점이 크지만 사각지역을 처리하지 못하는 단점으로 인해 정확도가 많이 떨어지고, 초음파센서 또한 비용적으로 유리하지만 다수의 센서 사용 시 간섭효과로 인해 정밀도가 떨어짐을 볼 수 있고 처리시간이 늦은 단점이 있다. 스테레오 비전 시스템은 스테레오 카메라의 성능에 따른 정확도와 처리시간의 차이가 많이 발생하는데, 고가일수록 정확도와 처리시간이 짧아진다. 하지만 스테레오 비전을 사용할 경우 입체적인 표현이 가능하고 로봇의 이동을 최소화 시킬 수 있는 장점이 있다. 레이저센서는 가장 정확한 정밀도를 갖지만 비용이 고가이며 단일 센서를 사용할 경우 틸팅 시스템이나 회전을 시켜서 측정해야 하므로 처리시간이 길어진다. 본 논문에서 제안한 방법은 영상처리를 통해 단일 센서를 사용한 시스템의 정밀도의 단점을 보완하였고 비용적인 측면에서 유리함을 가진다고 할 수 있다. 하지만 영상처리 시스템은 실험환경 변화에 민감하기 때문에 조명이나 움직이는 물체 발견 시에는 오차가 증가할 수 있다.

표 1. 타 시스템과의 성능 비교

Table 1. Comparison with other system about performance.

	Accuracy (m)	Computing Time(ms)	Cost
PSD	0.55	15	Very Low
초음파센서	0.21	150	Low
스테레오 비전	0.1	120	High
레이저센서(1unit)	0.01	95	High
PSD+단일카메라	0.07	85	Middle

## VI. 결 론

본 논문에서는 거리측정센서인 PSD 센서와 단일 영

상으로부터 특징점을 추출하여 환경맵을 작성하고 로봇의 위치인식이 가능한 SLAM 시스템을 구현하였다.

초음파센서에 비해 외란이 적고 LRF(laser range finder)에 비해 비용이 저렴한 PSD 센서를 사용하였고, 단일 카메라를 사용하여 기존 스테레오 비전 시스템의 긴 처리시간 및 높은 사양을 대처하였다. 영상위치획득 알고리즘과 확률분포 SLAM을 적용하여 지도생성과 위치인식 시스템을 설계하였으며, LABVIEW 7.0과 VISION BUILDER 3.0을 사용하여 제어프로그램과 영상처리시스템을 제작하였다. 로봇 회전 및 이동 실험을 통해 지도획득 성능 및 위치인식 시스템을 검증하였다. 이를 통해 경제적이면서 효율적인 SLAM 시스템을 구현할 수 있음을 보였다.

추후 과제로는 기존의 2D 맵을 3차원으로 구현할 수 있는 시스템 및 알고리즘 구현과 활동영역확대에 따른 시스템의 처리시간을 줄이는 방안과 지도생성의 효율성 증대가 되겠다.

## 참 고 문 헌

- [1] 이희성, 김은태, “외향 기반 환경 인식을 사용한 이동 로봇의 위치인식 알고리즘,” 전자공학회논문지, 제41권 CI편, 제 6호, 47-53쪽, 2004년 11월
- [2] Hugh Durrant-Whyte, Tim Bailey, “Simultaneous Localization and Mapping: Part 1,” *IEEE Robotics and Automation Magazine*, pp 99-108, June 2006.
- [3] 구자일, 홍준표, 이원석, “동기 구동형 이동로봇의 자율주행을 위한 위치측정과 경로계획에 관한 연구,” 전자공학회논문지, 제43권 IE편, 제 1호, 59-66쪽, 2006년 3월
- [4] J. Crowley, “World modeling and position estimation for a mobile robot using ultra-sonic ranging,” *Proc. IEEE Int. Conf. Robot. Automat.*, 1989, pp. 674-681, AZ, USA, May 1989.
- [5] Andreas Nuchater, Hartmut Surmann, and Sebastial Thrun, “6D SLAM with Application in Autonomous Mine Mapping,” *Proc. IEEE 2004 Int. conf. Robo. Automat.*, New Orleans, USA, April 2004.
- [6] Stephen Se, David Lowe, Jim Little, “Mobile Robot Localization and Mapping with Uncertainty using Scale-Invariant visual Landmarks,” *The Int. Journal on Robotics Research*, Vol. 21, No. 8, pp. 735-758, August 2002.
- [7] Andrew J. Davison, Ian D. Reid, Nicholas D.

- Molton and Olivier Stasse, "MonoSLAM: Real-Time Single Camera SLAM," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 29, No. 6, pp. 1052-1067, June 2007.
- [8] David G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journals of Computer Vision*, Vol. 60, No. 2, pp. 91-110, November 2004.
- [9] J. Diebel, K. Reutersward, S. Thrun, J. Davis, G. Gupta, "Simultaneous Localization and Mapping with Active Stereo Vision," *Proc. of 2004 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 3436-3443, Sendai, Japan, September 2004.
- [10] 진태석, 이장명, "단일 초음파 센서모듈을 이용한 이동로봇의 위치추정 및 주행," 전자공학회논문지, 제42권 SC편, 제 2호, 1-10쪽, 2005년 3월
- [11] Gordon Chung, Alexander Zelinsky, "Real-Time Visual Behaviours for Navigating a Mobile Robot," *Proc. IROS 96*, pp 973-980, Osaka, Japan, November 1996.
- [12] S. Thrun, "Probabilistic algorithms in robotics," *Technical Report CMU-CS-00-126*, Carnegie Mellon Univ., Computer Science Department, Pittsburgh, PA, 2000.
- [13] Greg Welch, Gray Bishop, "An Introduction to the Kalman Filter," SIGGRAPH, 2001.
- [14] R.G. Brown, "Introduction to Random Signals and Applied Kalman Filtering," *Wiley Press*.

---

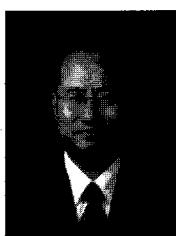
### 소 개

---



유 성 구(정회원)  
 2003년 전북대학교 제어계측  
 공학과 학사졸업.  
 2005년 전북대학교 제어계측  
 공학과 석사졸업.  
 2008년 현재 전북대학교 제어계측  
 공학과 박사과정

<주관심분야 : Robotics, 인공지능, 제어시스템>



정 길 도(정회원)  
 1984년 Oregon State University  
 기계공학 학사졸업.  
 1986년 Georgia Institute of  
 Technology 기계공학  
 석사졸업.  
 1992년 Texas A&M University  
 기계공학 박사 졸업.  
 2008년 현재 전북대학교 전자정보 교수  
 <주관심분야 : Time-Delay, Robotics, 인공지능,  
 Web 기술>