

# 로봇 위치 예측에 기반을 둔 ICP 알고리즘을 이용한 지도 작성

노성우\* · 김태균\*\* · 고낙용\*\*\*

Map Building Using ICP Algorithm based a Robot Position Prediction

Sung-Woo Noh\* · Tae-Gyun Kim\*\* · Nak-Yong Ko\*\*\*

## 요약

본 논문에서는 로봇 위치 예측 기반을 둔 ICP 알고리즘을 이용한 지도를 작성한다. 제안한 방법은 데드레크닝으로 로봇 위치를 예측하고 ICP 알고리즘으로 지도를 작성 한다. 기존 방법은 기준데이터와 새 데이터의 센서 값만을 이용하여 로봇의 위치와 지도를 작성한다. 기존 방법은 현재 데이터와 기준 값과의 간격의 차이가 조금만 멀어져도 보정하기가 어렵다. 하지만 제안한 방법으로 지도를 작성할 경우에는 지도의 틀어진 정도가 기존 방법으로는 지도를 보정 할 수 없지만 제안한 방법은 지도 보정을 할 수 있음을 실제 실험을 통해 나타내었다.

## ABSTRACT

This paper proposes a map building using the ICP algorithm based robot localization prediction. Proposed method predicts a robot location to dead reckoning, makes a map in the ICP algorithm. Existing method makes a map building and robot position using a sensor value of reference data and current data. In this case, a large interval of the difference of the reference data and the current data is difficult to compensate. The proposed method can map correction through practical experiments.

## 키워드

ICP(Iterative Closest Point), Map building, Scan matching, Dead reckoning, Mobile robot  
지도작성, 스캔 매칭, 데드레크닝, 이동로봇

## 1. 서론

자율주행 이동로봇은 다양한 외부 환경 변화를 인지하여 자신의 위치를 파악하고 장애물을 회피하며 경로를 생성하여 목적지까지 스스로 주행해야 한다. 이를 위해서는 지도정보가 반드시 필요하다.

이동로봇은 주위에 관한 정보를 이미 알고 있거나 인식할 수 있는 형태의 정보로 제공되더라도 지도를 개선, 변형, 확장하는 것은 중요하다[1]. 예를 들어 책상, 의자

와 같은 가구, 장식품, 방문의 손잡이 등 로봇의 이동 및 서비스에 필요한 세세한 공간 정보는 설계도에 반영하기 어렵고 대상물의 위치와 방향이 수시로 바뀔 수 있다. 따라서 이동로봇은 작업환경에 대한 정보를 로봇 스스로 알아내고 지도 작성을 수행해야 한다.

기존의 이동로봇을 위한 지도 작성은 센서와 데이터저장 방법에 따라 점유 격자지도(Occupancy grid map), 특이점 지도(Feature based map), 위상지도(Topological map), 스캔 매칭지도(Scan matching

\* 조선대학교 정보통신공학과(nswking0212@naver.com)

\*\* 조선대학교 제어계측공학과(ktg9114@naver.com)

\*\*\* 교신저자 : 조선대학교 제어계측로봇공학과(nyko@chosun.ac.kr)

접수일자 : 2013. 01. 11

심사(수정)일자 : 2012. 03. 25

게재확정일자 : 2013. 04. 22

map)로 나눌 수 있다[2-8].

본 논문에서는 로봇 위치 예측 기반을 하고 스캔매칭 방법인 ICP 알고리즘을 이용하여 지도 작성 방법을 제안한다. 이 방법은 로봇이 이동하기 전의 위치에서 스캔한 기준 데이터는 정확하다고 가정한다. 그리고 로봇을 이동시킨 후, 스캔한 새 데이터는 기준 데이터와 비교하여 로봇위치를 보정하고 동시에 지도를 작성한다. 다시 말하면, 스캔 매칭은 모션모델의 불확실성을 센서모델들의 매칭기법으로 로봇의 회전 오차, 직진 오차를 구함으로써 로봇의 위치 오차를 보정하고 지도를 작성하게 된다.

본 논문 구성은 다음과 같다. 2장에서는 지도 작성을 위한 기준 방법과 제안한 방법을 설명한다. 3장에서는 실제 이동로봇을 이용하여 레이저 센서에서 획득된 데이터로 데드레크닝, 기존 ICP 방법, 제안한 ICP 방법을 실제 환경에서 실험한다. 그리고 4장에서 결론을 내리고자 한다.

## II. 지도 작성을 위한 스캔매칭 방법

스캔 매칭은 현재의 새로운 거리 데이터와 과거에 측정된 기준 거리 데이터(또는 지도로부터 주어진 데이터)를 서로 비교한다. 그림 1은 점과 점에 대한 스캔매칭의 결과이다. 빨간색 데이터는 기준 데이터이고

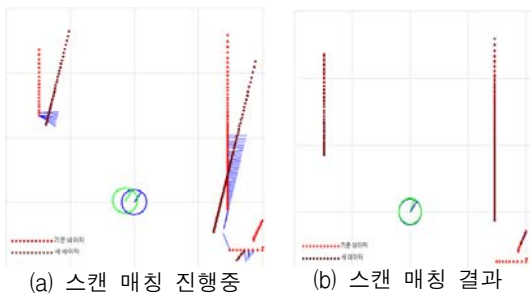


그림 1. 스캔매칭 기본 개념  
Fig. 1 Scan matching basic concepts

파란색 로봇은 기준로봇이다. 그리고 갈색 점 데이터는 새 데이터이고 녹색 로봇은 갱신해야 할 로봇이다. (a)는 스캔 매칭 진행 상태를 나타낸다. 기준데이터는 새 데이터와 비교하고 로봇의 위치 및 점 데이터를

보정하게 된다. (b)는 스캔매칭 결과로서, 로봇과 데이터가 보정된 단계이다.

### 2.1 ICP 알고리즘 방법

ICP 알고리즘은 영상처리에서 사용되던 기술로서, 이를 주행 알고리즘에 적용하면서 가장 대표적인 점과 점 스캔 매칭 알고리즘으로 발전되었다. ICP는 거리 센서에 의해서 얻은 새 데이터와 기준 데이터에서 유클리드 거리가 가장 가까운 데이터들을 일치 쌍을 계산한다. 그리고 FLu가 제안한 closed-form solution 방법은 현재 로봇의 정확한 위치를 보정하고 지도를 작성한다[10].

기준데이터  $q_i$ , 입력 데이터(새 데이터)  $p_i$ , 기준데이터의 법선 벡터  $n_i$  를 알고 있다면, 입력 데이터가 모델 데이터에 에러  $e$ 를 최소화 하는 회전행렬  $R$ 과 이동 벡터 값  $t$ 를 식 1과 같이 구할 수 있다[12].

$$e = \sum_{i=1}^n ((Rp_i + t - q_i) \cdot n_i)^2$$

$p_i = (p_{i,x}, p_{i,y})^T$ : 입력데이터  
 $q_i = (q_{i,x}, q_{i,y})^T$ : 기준데이터  
 $t = (x, y)$ : 이동벡터  
 $R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \approx \begin{bmatrix} 1 & -\theta \\ \theta & 1 \end{bmatrix}$ : 회전행렬  
 $n_i = (n_{i,x}, n_{i,y})^T$ : 기준데이터의법선벡터

식 1을 정리하면 식2와 같이 정리된다.

$$e = \sum_{i=1}^n \left( \left( \begin{bmatrix} 1 & -\theta \\ \theta & 1 \end{bmatrix} p_i + \begin{bmatrix} x \\ y \end{bmatrix} - q_i \right) \cdot n_i \right)^2$$

$$= \sum_{i=1}^n \left( (p_i - q_i) \cdot n_i + t \cdot n_i + \begin{bmatrix} -p_{i,y} \\ p_{i,x} \end{bmatrix} \cdot n_i \theta \right)^2$$

식 2에서 계산을 간편하게 하기위해 치환을 이용하여 식 3과 같이 만든다.

$$e = \sum_{i=1}^n (\alpha_i + t \cdot n_i - \beta_i \theta)^2$$

$\alpha_i = (p_i - q_i) \cdot n_i, \beta_i = (-p_{i,y}, p_{i,x})^T \cdot n_i$

식 3을 다시 풀어보면, 식 4와 같이 계산된다.

$$e = \sum_{i=1}^n (\alpha_i + t \cdot n_i - \beta_i \theta)^2 \quad (4)$$

$$\because \alpha_i = (p_i - q_i) \cdot n_i, \beta_i = (-p_{i,y}, p_{i,x})^T \cdot n_i$$

x, y,  $\theta$ 에 대한 에러의 최소값을 구하기 위하여 식 4를 식 5와 같이 e를 x, y,  $\theta$ 로 편미분 한다.

$$\begin{aligned} \frac{\partial e}{\partial x} &= \sum_{i=1}^n 2(\alpha_i + xn_{i,x} + yn_{i,x} + \beta_i \theta) \cdot n_{i,x} = 0 \\ \frac{\partial e}{\partial y} &= \sum_{i=1}^n 2(\alpha_i + xn_{i,x} + yn_{i,y} + \beta_i \theta) \cdot n_{i,y} = 0 \\ \frac{\partial e}{\partial \theta} &= \sum_{i=1}^n 2(\alpha_i + xn_{i,x} + yn_{i,y} + \beta_i \theta) \cdot \beta_i = 0 \end{aligned} \quad (5)$$

식 5를 행렬 형태로 변환하여 나타내면 식 6와 같이 바뀔 수 있다.

$$\sum_{i=1}^n \begin{bmatrix} n_{i,x}^2 & n_{i,x}n_{i,y} & \beta_i n_{i,x} \\ n_{i,x}n_{i,y} & n_{i,y}^2 & \beta_i n_{i,y} \\ \beta_i n_{i,x} & \beta_i n_{i,y} & \beta_i^2 \end{bmatrix} \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} = \sum_{i=1}^n \begin{bmatrix} \alpha_i n_{i,x} \\ \alpha_i n_{i,y} \\ \alpha_i \beta_i \end{bmatrix} \quad (6)$$

선형 행렬 식 6은  $AX = b$  형태로 표현가능하며,  $X=A^{-1}b$ 와 같이 나타내어  $X = (x, y, \theta)$ 를 구할 수 있다.

그림 2는 ICP 알고리즘의 의사코드이다.

```

=====
Algorithm ICP
=====
1: for i=0 to max Iterations do
2:   for all dj ∈ D do
3:     find the closest point within a range dmax in the set M for point dj
4:   end for
5:   Calculate transformation (R,t) that minimizes the error function
6:   Apply the transformation found in previous step to the data set D
7:   Compute the difference of the quadratic error before
   and after the application of the transformation
8:   If this difference falls below a threshold ε, terminate.
9: end for
=====
    
```

그림 2. ICP 의사코드  
Fig. 2 ICP Pseudo-code

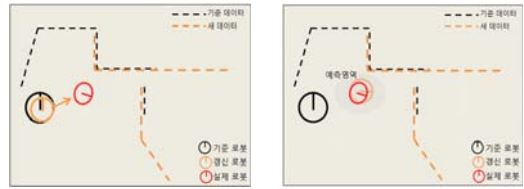
그림 2에서의 Line 5는 ICP알고리즘을 이용하여 에러 최소값을 계산하고 Line 8은 직진오차와 회전오차가 임계값보다 크면 스캔 매칭을 다시 실행하고, 임계값 보다 작으면 수렴하였다고 판단하여 스캔매칭을

종료한다. ICP를 이용한 스캔매칭은 N개의 기준데이터와 N개의 새 데이터를 일대일로 비교해야한다. 이때, 계산 복잡도는  $O(N^2)$ 이 되어 많은 수행시간이 필요하다. 따라서, 이러한 단점을 극복하기 위해 Kd tree 검색 방법을 이용한다[11].

## 2.2 예측 위치에 기반한 ICP

본 절에서는 데드레코닝으로 로봇 위치를 예측하고 ICP 알고리즘 기반으로 하는 이동로봇 지도 작성 방법을 다룬다. 기존 방법은 기준데이터와 새 데이터의 센서 데이터 값만을 이용하여 로봇의 위치 및 지도를 작성하였다. 이 경우는 로봇의 위치데이터 예측값이 없다. 따라서, 로봇의 위치는 기준 값과의 간격의 차이가 조금만 멀어져도 보정하기가 어렵다.

제안한 방법은 데드레코닝 정보를 이용하여 예측된 위치를 기반으로 새 데이터와 기준 데이터를 비교하여 로봇의 위치를 보정하면서 지도를 작성한다. 이는 기존 방법보다 효율적으로 지도를 작성할 수 있다.



(a) 기존 방법 (b) 제안한 방법

그림 3. 기존 방법과 제안한 방법 비교  
Fig. 3 Comparison of the existing method and proposed method

그림 3은 기존 방법과 제안한 방법 비교를 나타낸다. (a)의 방법은 기존 데이터와 새 데이터가 비교 할 때 단지 센서 데이터만을 이용하여 로봇의 위치 및 지도를 작성한다. 이 경우에는 실제 로봇의 위치가 기준 로봇과 멀리 떨어져 있을 때에는 갱신로봇의 위치 보정이 어렵다. (b)의 방법은 갱신 로봇의 위치를 미리 예측을 하여 실제 로봇의 위치에 접근 시켜 놓고 기준 데이터와 새 데이터를 비교한다. 이 경우에는 기준로봇과 실제 로봇과 위치 차이가 있더라도 갱신로봇은 실제로봇 주변에 있게 되어 위치 보정 및 지도를 작성 할 수 있다.

그림 4는 제안한 방법으로 지도 작성을 하기 위한

의사코드이다.

```

=====
Predictive ICP( $X_{t-1}, u_t, z_{t-1}, z_t$ )
=====
1:  $\bar{X}_t = \text{Prediction}(X_{t-1}, u_t)$ 
2:  $X_t = \text{ICP Algorithm}(\bar{X}_t, z_{t-1}, z_t)$ 
3:  $M = (X_t, z_t)$ 
4: return M
=====
    
```

그림 4. 제안한 방법의 의사코드  
Fig. 4 Pseudo-code of proposed method

Line1은 t 이전 시점  $X_{t-1}$ 로봇 위치정보에서 속도정보  $u_t$ 를 이용하여 갱신할 로봇위치  $X_t$ 를 예측한다. Line2는 예측된 로봇  $X_t$ , 기준 데이터  $z_{t-1}$ 와 현재 새 데이터  $z_t$ 를 ICP 알고리즘을 통해 비교하여  $X_t$  로봇 위치를 찾는다. Line3은 현재 로봇 위치  $X_t$ , 새 데이터  $z_t$ 를 이용하여 지도 작성을 한다.

그림 5는 차륜형 방식의 로봇으로 2개의 구동 바퀴와 1개의 지지용 바퀴로 동작하는 이동 로봇이다. 이러한 방식의 로봇은 각각의 바퀴들이 개별적으로 동작하여 이동로봇의 위치( $x, y$ )와 진행 각도( $\theta$ )가 시간에 따라 달라진다. 여기서,  $W_L$ 과  $W_R$ 은 왼쪽 바퀴와 오른쪽 바퀴의 회전 속도 정보,  $\theta$ 는 로봇의 방향을 나타내는 헤딩각,  $r$ 은 바퀴의 반지름, 그리고  $d$ 는 왼쪽 바퀴와 오른쪽 바퀴 사이의 거리를 나타낸다.

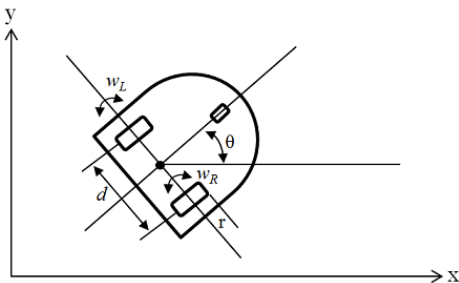


그림 5. 차륜형 이동로봇 구동방식  
Fig. 5 Driven approach of wheeled mobile robot

차륜형 이동로봇에 대한 속도 정보는 식 7과 같이 그림 5의 파라미터들을 이용하여 정의될 수 있다[13]. 여기서  $v$ 는 로봇의 무게 중심에서의 직진속도,  $w$ 는 로봇의 회전 속도를 의미한다.

$$v = \frac{r(w_R + w_L)}{2}, w = \frac{r(w_R - w_L)}{d} \quad (7)$$

그림 6은 그림 4 Line(1)에 대한 차륜 구동 방식의 위치예측 의사코드이다. 이동로봇의 직진속도  $v$ 와 회전속도  $w$ 를 알 수 있을 때 t시간의 이동로봇 위치( $x_t, y_t, \theta_t$ )는 다음과 같이 예측할 수 있다[14].

```

=====
Prediction Pose( $X_{t-1}, v, \omega$ )
=====
1:  $x_t = x_{t-1} - \frac{v}{\omega} \sin \theta_{t-1} + \frac{v}{\omega} \sin(\theta_{t-1} + \omega \Delta t)$ 
2:  $y_t = y_{t-1} + \frac{v}{\omega} \cos \theta_{t-1} - \frac{v}{\omega} \cos(\theta_{t-1} + \omega \Delta t)$ 
3:  $\theta_t = \theta_{t-1} + \omega \Delta t$ 
4: return  $\bar{X}_t = (x_t, y_t, \theta_t)$ 
=====
    
```

그림 6. 차륜 구동방식의 위치 예측 의사코드  
Fig. 6 Localization Predict Pseudo-code of Wheeled Mobile Robot

### III. 실험 및 고찰

본 실험에서는 3가지 형태로 실험을 진행하였다. 실험 1은 데드레크닝 방법과 ICP 알고리즘을 적용한 지도 작성 방법을 비교 분석하고 실험2는 데드레크닝, 기존 ICP방법, 제안한 방법을 비교 분석한다. 실험3은 기존 ICP방법, 제안한 방법으로 속도에 따른 지도 작성을 분석한다.

표 1은 레드원테크놀로지(주)에서 개발된 실험에서 사용한 차륜형 이동 로봇이다[15]. 아래 그림은 이동 로봇의 제원을 나타내었다. 지도를 작성하기 위한 거리 센서는 Sick社에서 개발한 LMS-511이다.

표 1. 이동로봇 제원  
Table 1. Mobile robot specifications

Item	Specification
Dimension	(L)560mm × (W)450mm × (H)320mm
Weight	21kg
Drive	2-wheel drive rear balancing caster
Payload	Max. 40Kg
Steering	Differential (2 Wheel)
Translate speed	Max. 1.2m/s
Continuous run time	2 ~ 5 hours
Motor	60W BLDC motor, Max. 3,000
Sensor	Front IR-PSD Sensor, Range : 0 ~ 160cm
Controller(optional)	Note-book, SBC, IPC, PC
Operating system	Windows OS (2000, XP)
Control languages	Visual C++, Visual Basic, IEC61131-3 PLC Languages
Network type	Real time Industrial Ethernet for Motion Controller
Middleware	MSRDS, RTC(OMG Robot Standard), iRSP, MATLAB(optional)



### 3.1 실험(1)

본 절에서는 데드레크닝과 ICP 알고리즘을 이용한 지도작성 실험이다. 본 실험환경은 그림 7에서 보인 조선대학교 전자정보공과대학 6122호 자율로봇 실험실에서 수행하였다.



그림 7. 실험 환경(1)  
Fig. 7 Experimental environment(1)

실험환경은 실제 도면을 이용하여 지도를 작성 할 수 없기 때문에 정량적으로 표현 하지는 않는다. 단, 데드레크닝과 ICP 방법을 사용할 때, 지도 작성 표현이 어떻게 이루어지는지 보는 실험이다.

본 환경에서 이동로봇은 1회 왕복하여 지도를 작성 하였다. 그림 8의 격자의 단위는 가로, 세로 1m이다. 전체 지도 작성 환경의 크기는 가로(8m), 세로(8m)이다.

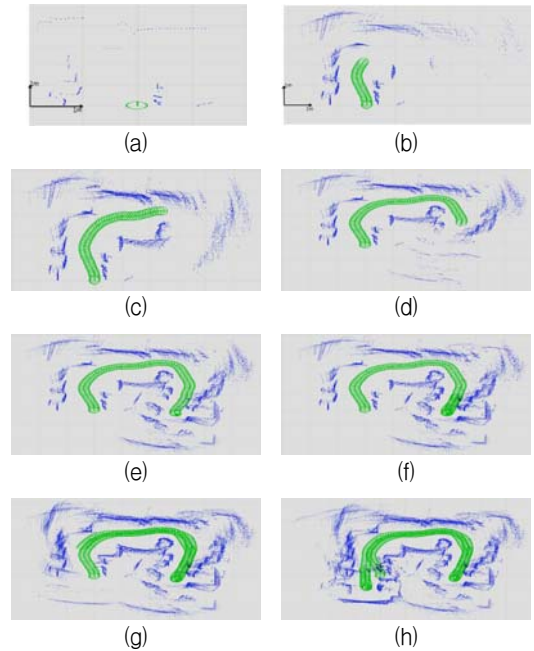
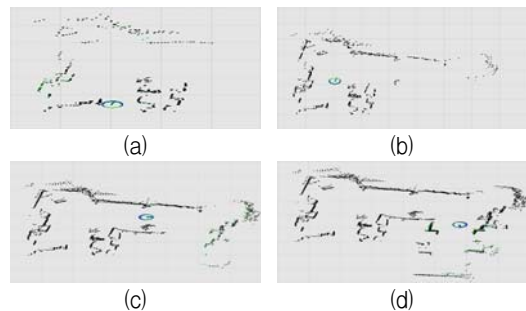


그림 8. 데드레크닝을 이용한 지도작성 실험결과  
Fig. 8 Map building experimental results using dead reckoning

그림 8은 데드레크닝을 이용한 방법이다. 데드레크닝을 이용한 방법은 (a)~(h)와 같이 시간이 지날수록 지도가 틀어지는 현상이 발생한다. 이유는 엔코더 신호의 부정확성, 바퀴의 마찰, 미끄러짐 등 다양한 현상으로 인해 로봇의 위치가 부정확한 상태에서 지도를 만들기 때문이다.

그림 9는 ICP 알고리즘을 적용하여 지도를 작성한다. 아래 그림에서 녹색으로 보이는 로봇은 새 데이터(입력 데이터)를 나타낸다. 새 데이터는 이전 기준데이터와 비교하여 로봇 위치를 보정한 후, 지도를 작성한다.



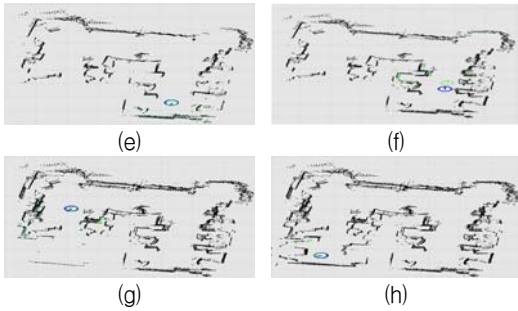


그림 9. ICP알고리즘을 이용한 지도작성 실험결과  
Fig. 9 Map building experimental results using ICP algorithm

3.2 실험(2)

실험 2에서는 지도 작성을 정량적으로 표현하기 위해 직접 실험환경을 구성하였다. 그림 10은 실험을 수행한 조선대학교 전자정보공과대학 6103호 컴퓨터 실험실 환경이다. (a)는 실험 공간을 나타내었고 (b)는 실험공간의 평면도이다. 실험 공간 환경의 격자는 1m 이고 총 크기는 가로(4m), 세로(5m)이다. 본 실험에서는 파란색 화살표로 방향으로 1번, 빨간색 화살표 방향으로 1번 로봇을 이동하면서 수행하였다.

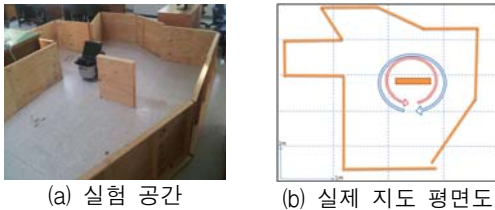


그림 10. 실험 환경(2)  
Fig. 10 Experimental Environment(2)

실험 방법은 데드레크닝, ICP 알고리즘 방법, 제안한 방법을 수행하였다. 그림 11은 각각의 방법에 대한 실험 결과를 나타낸다. 그림에서 보면, (a)의 방법은 데드레크닝으로 지도를 작성한 경우이다. 파란색은 지도 점 데이터를 나타내고 실제 환경은 갈색으로 나타내었다. 데드레크닝 방법은 실제 환경과 비교해보면 오차 범위가 크다. (b)의 방법은 ICP 알고리즘만을 이용하여 나타내었다. 검은색은 지도 점 데이터를 나타낸다. 이 경우는 (a)방법보다 실제 환경과 유사하게

보인다. 하지만 실제 환경과는 차이가 있다. (c)는 제안한 방법(데드레크닝+ICP 알고리즘)을 이용하여 나타내었다. (c)의 경우에는 (a), (b)의 방법을 적용했을 때 보다 실제 환경과 유사한 지도를 작성한다.

표 2는 그림 11 (b), (c)의 E1~E11부분의 실제 환경과 지도 작성 할 때 매칭 되어야 할 최대 오차거리를 나타내었다.

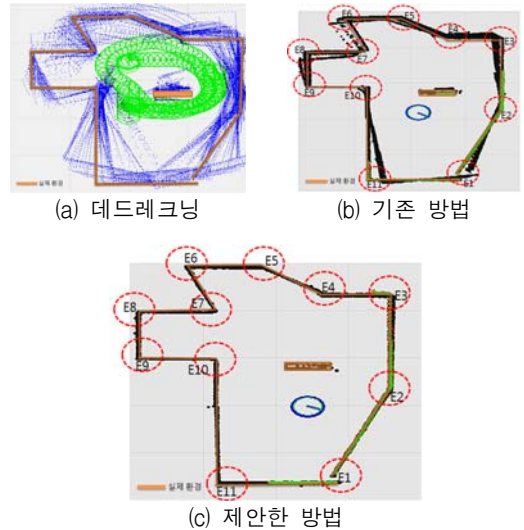


그림 11. 지도 작성 방법에 따른 실험 결과  
Fig. 11 Experimentation result according to map building method

표 2. 지도 작성 오차 거리  
Table 2. Map building error distance

	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11
기존 방법	25	8	27	12	8	37	14	17	5	12	30
제안한 방법	8	3	7	3	2	6	4	3	3	3	3

(cm)

3.3 실험(3)

본 실험에서는 로봇 이동 속도에 따른 지도 작성 성능을 비교한다. 실험 환경은 실험 2의 작업 공간 환경과 동일하다. 본 실험에서는 로봇의 직진속도 0.4m/s, 회전속도 0.6rad/s로 구성하고 기존에 사용된 ICP 방법과 제안한 ICP 방법을 비교 실험한다. 그림 12는 기존

ICP 방법을 이용한 실험 결과이다. 빨간색은 기준데이터를 나타내고 파란색 로봇은 갱신하기 전 로봇을 나타낸다. 그리고 녹색은 새 데이터이고 녹색 로봇은 보정을 하기위한 로봇이다. 검은색은 만들어진 지도이다. 그림 12 (a)~(c)는 기존 ICP 방법을 이용한 처리순서이다. (a)는 실제 로봇이 이동하여 새 데이터가 변화고 있는 화면이고 (b)는 ICP 방법을 이용하여 녹색 로봇을 보정하려는 화면이다. (c)는 기준데이터와 새 데이터를 비교하여 보정된 결과이다. (a)~(c)의 처리 과정을 살펴보면, 기준 데이터와 새 데이터간의 매칭이 불일치되어 있다.

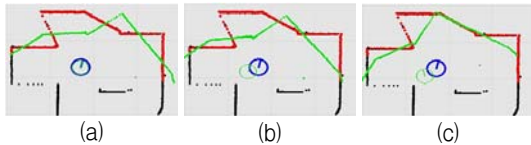


그림 12. 기존 ICP 방법을 이용한 이동로봇 속도에 따른 지도 작성 결과

Fig. 12 Map building result according to mobile robot velocity using existing ICP method

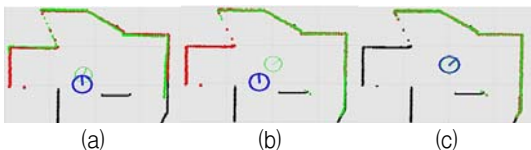


그림 13. 제안한 방법을 이용한 이동로봇 속도에 따른 지도 작성 결과

Fig. 13 Map building result according to mobile robot velocity using proposed method

그림 13 (a)~(c)는 제안한 ICP 방법을 이용한 처리순서이다. (a)→(b)→(c)와 같은 경우에는 기준데이터와 새 데이터 간의 위치 보정이 이루어져 지도를 작성할 수 있었다.

실험을 통해 알 수 있는 결과로서, 기존의 방법은 이동로봇 속도가 빠를 경우나 기준데이터와 새 데이터간의 간격이 커질 경우 로봇의 보정 및 지도를 작성할 수 없음을 실험을 통해 알 수 있다. 하지만 제안된 방법은 이동속도나 비교할 데이터 간격이 커지더라도 로봇의 위치 보정 및 지도를 작성 할 수 있었다.

#### IV. 결론

본 논문에서는 로봇 위치 예측 기반을 둔 ICP 알고리즘을 이용한 지도를 작성하였다. 기존 방법인 기준데이터와 새 데이터의 센서 값만을 이용하여 로봇의 위치 및 지도를 작성하였을 때, 로봇의 위치와 기준값과의 간격의 차이가 조금만 멀어져도 보정하기가 어렵다. 하지만 제안한 방법을 이용하였을 경우는 데드레크닝 정보를 이용하여 예측된 위치를 기반으로 새 데이터와 기준 데이터를 비교하여 로봇의 위치를 보정하면서 지도를 작성하였기 때문에 기존 방법보다 지도 정확성이 향상되었고 수행 시간도 단축할 수 있었다.

향후 과제는 로봇 이동 속도에 따른 지도 작성 가능 유무를 정량적으로 표현 가능토록 수치화한다. 또한, 기존 ICP 알고리즘에서 기준데이터와 새 데이터를 비교할 수 있도록 제안된 Kd-tree 방법을 시간 단축을 위해 새로운 방법을 고안할 예정이다.

#### 감사의 글

본 연구는 교육과학기술부, 한국연구재단의 2010년 지역혁신인력양성사업(과제명 : 로봇의 자율주행 요소 기술 상용화 및 인력양성, 과제관리번호 : 2010-04-대-01-016)의 지원에 의해 이루어짐

#### 참고 문헌

- [1] T. Tsubouch, "Nowadays trends in map generation for mobile robots," IEEE International conference on Intelligents Robots and Systems, pp. 828-833, 1996.
- [2] J. Borenstein and L. Feng, "Measurement and Correction of Systematic Odometry Errors in Mobile Robots," IEEE Trans. on Robotics and Automation, Vol. 12, No. 6, pp. 869-880, 1996.
- [3] K. Lee, C. Chung, and W. Chung, "Accurate Calibration of Kinematic Parameters for Two Wheel Differential Mobile Robots," Journal of Mechanical Science and Technology, Vol. 25, No. 6, pp. 1603-1611, 2011.
- [4] 권용관, "가상 탄성력을 이용한 자율이동로봇 경로생성 방법", 한국전자통신학회논문지, 8권, 2

호, pp. 149-157, 2013.

[5] 문용선, 노상현, 조광훈, 배영철, "정밀 GPS 기반의 실외환경에서의 로봇 위치 추정 및 openRTM을 이용한 모니터링", 한국전자통신학회논문지, 7권, 2호, pp. 425-432, 2012.

[6] 권태범, 송재복, 강신천, "실외 이동로봇의 고도 지도 기반 위치인식을 위한 고도관성모멘트 추출 및 정합", 제어·로봇·시스템학회논문지, 15권, 2호, pp. 203-210, 2009.

[7] 윤강섭, "초음파 위성 시스템을 위한 개선된 위치추정 알고리즘", 한국전자통신학회논문지, 6권, 5호, pp. 775-781, 2011.

[8] G. Casella and R. L. Berger, Statistical Inference, Duxbuty, 2002.

[9] F.Lu, E.Milions, "Robot pose estimation in unknown environments by matching 2d range scan.", Journal of Intelligent and Robotic Systems, Vol. 18, pp. 249-275, 1992.

[10] Moore, A. W., "An Introductory Tutorial on Kd- Trees," Extract from Efficient Memory based Learning for Robot Control(Technical Report 209), Computer Laboratory, University of Cambridge, 1991.

[11] Chen, Y. and Medioni, G. "Object Modeling by Registration of Multiple Range Images," Proc. IEEE Conf. on Robotics and Automation, 1991.

[12] 김태균, "유비쿼터스 센서 환경에서의 이동로봇 위치추정", 조선대학교 석사학위논문, 2009.

[13] S. Thrun, W. Burgard, and D. Fox, "Probabilistic Robotics", MIT Press, 2005.

[14] <http://www.redone-technologies.com/>

저자 소개



**노성우(Sung-Woo Noh)**

2007년 조선대학교 제어계측공학과 졸업(공학사)  
 2010년 조선대학교 대학원 제어계측공학과 졸업(공학석사)

2010년~현재 조선대학교 대학원 정보통신공학과 박사과정

※ 관심분야 : 이동로봇, 경로계획, Microprocessor



**김태균(Tae-Gyun Kim)**

2007년 조선대학교 제어계측공학과 졸업(공학사)

2009년 조선대학교 대학원 제어계측공학과 졸업(공학석사)

2009년~현재 조선대학교 대학원 제어계측학과 공학박사과정

※ 관심분야 : 이동로봇, 수중로봇, 위치추정



**고낙용(Nak-Yong Ko)**

1985년 서울대학교 제어계측공학과 졸업(공학사)

1987년 서울대학교 대학원 제어계측공학과 졸업(공학석사)

1993년 서울대학교 대학원 제어계측공학과 졸업(공학박사)

1997~1998, 2004~2005 미국 Carnegie Mellon Univ. Visiting research scientist

1992년~현재 조선대학교 제어계측로봇공학과 교수

※ 관심분야 : 지상로봇과 수중로봇의 자율주행