

3차원 공간 맵핑을 통한 로봇의 경로 구현

Implementation of Path Finding Method using 3D Mapping for Autonomous Robotic

손은호, 김영철, 정길도*

(Eun-Ho Son, Young-Chul Kim, and Kil-To Chong)

Abstract : Path finding is a key element in the navigation of a mobile robot. To find a path, robot should know their position exactly, since the position error exposes a robot to many dangerous conditions. It could make a robot move to a wrong direction so that it may have damage by collision by the surrounding obstacles. We propose a method obtaining an accurate robot position. The localization of a mobile robot in its working environment performs by using a vision system and Virtual Reality Modeling Language(VRML). The robot identifies landmarks located in the environment. An image processing and neural network pattern matching techniques have been applied to find location of the robot. After the self-positioning procedure, the 2-D scene of the vision is overlaid onto a VRML scene. This paper describes how to realize the self-positioning, and shows the overlay between the 2-D and VRML scenes. The suggested method defines a robot's path successfully. An experiment using the suggested algorithm apply to a mobile robot has been performed and the result shows a good path tracking.

Keyword : computer vision, path finding, autonomous robot localization, back-propagation, VRML

I. 서론

자율이동로봇은 서비스로봇과 더불어 활발히 연구되어지고 있다. 이동로봇은 근래에는 가정용, 사무용, 산업용, 의료용 등 많은 분야에서 응용되어지고 있다. 특히 장애물이 많은 실내에서는 로봇이 스스로 실내구조나 장애물을 파악해 충돌을 회피하고 최적경로를 생성하여 안전하게 임무를 수행할 수 있는 능력을 갖추고 있어야 한다. 먼저 이러한 작업환경 내에서 보다 정확하고 안전하게 임무를 수행하기 위해서는 그 자신의 위치파악이 무엇보다도 선행되어야 하며 그 동안 많은 연구들을 통해 로봇의 효율적인 측위 기법이 개발되어 왔다.

이동로봇의 경로계획에 대한 연구는 크게 로드맵(roadmap) 방식, 셀 분할(cell decomposition)방식, 퍼텐셜 필드(potential field) 방식 등의 연구가 진행되고 있다. 그러나 이 같은 경로계획 방법들은 로봇의 위치정보 에러로 인해 많은 문제점들을 갖고 있으며, 이와 같은 문제점 해결하기 위해 많은 연구가 진행되어지고 있다.

로봇의 위치추적 기법으로 비전시스템과 Landmark를 이용한 Triangulation 기법은 그 알고리즘의 구현이 쉽고 비교적 정확한 계산이 가능한 장점 때문에 오랫동안 이 분야에서 널리 사용되어 왔다[1,2]. Landmark란, 로봇이 쉽게 인지할 수 있는 작업 환경 내의 구조물들인 벽면의 수직선, 문고리, 천장의 배기구 등을 표시로서 활용하여 로봇의 측위를 계산하는 방식이다. 그러나 Landmark의 잘못된 인식이나 사이 각의 잘못된 측정에 의해 위치 계산결과는 항상

에러를 포함한다[3]. 이러한 에러를 보상하기 위해, 가장 인식이 좋은 최적의 Landmark만 선별하거나 수행된 자기측위 데이터를 통계학적인 분석에 의해 에러를 줄이는 방법들이 제안되어 왔다[2,4,5].

또한 가상현실 구현을 위한 VRML(Virtual Reality Modeling Language)은 Web3D 구현을 위한 3차원 가상 정보를 제공하기 위해 개발된 언어로, 이미 개발된 여러 툴들을 통해 쉽게 3차원 모델을 제작할 수 있으며, 3차원 모델 소스는 아스키코드로 구성되어 있고 그 분석이 쉽게 가능한 언어이다[8,9]. 이 VRML 3차원 영상은 로봇의 작업 환경을 모델링하여 로봇 작업환경의 정보를 모두 담고 있다. 그러므로 두 영상의 비교를 통해 위치 계산 결과의 에러 유무를 확인하고 쉽게 보정이 가능하다[6,7,10,11].

따라서 본 논문에서는 선형방정식의 최소제곱법을 적용한 linear position estimation 기법을 도입하여, CCD 카메라를 통해 입력된 비전 영상과 가상의 VRML(Virtual Reality Modeling Language) 3차원 영상의 매칭 방법을 적용하여, 이동로봇의 위치를 보정하고 로봇이 정확한 이동 경로를 찾아 임무를 수행하기 위한 효율적인 경로 계획법을 제안한다. 이를 위하여, 먼저 이동 로봇은 주위환경 내의 Landmark를 추출하고, 신경망 패턴인식 기법을 이용하여 위치정보를 갖는 Landmark를 정확하게 인식함으로써 로봇의 정확한 위치를 계산하게 된다. 또한, 실제 비전영상과 가상현실 공간 내의 이미지 정보를 비교 분석하여 이동로봇의 위치 인식오차를 최소한으로 하기 위하여 가상현실 구현 언어인 VRML(Virtual Reality Modeling Language)을 사용하여 가상공간을 구축하고, 입력 영상과 가상공간과의 오버레이를 통한 이동 로봇의 효율적인 위치보정 알고리즘을 제안함으로써, 이동로봇이 주행 중 장애물을 회피할 수 있는 능

* 책임저자(Corresponding Author)

논문접수 : 2007. 12. 27., 채택확정 : 2008. 1. 5.

손은호 : 전북대학교 제어계측공학과(yauchil@chonbuk.ac.kr)

김영철 : 군산대학교 기계공학부(kimyc@kunsan.ac.kr)

정길도 : 전북대학교 전자정보공학부(kitchong@chonbuk.ac.kr)

력을 갖도록 함으로 제안한 알고리즘의 유용성을 보인다.

본 논문의 구성은 다음과 같다. 2장에서는 위치판단을 위해 적용된 Localization 알고리즘에 대해 설명하고, 로봇 측위를 수행하는데 필요한 Landmark의 인식 및 영상처리 과정과 신경회로망을 이용한 패턴 인식 방법을 다룬다. 3장에서는 제안된 전체 시스템의 구성에 대해 알아보고, 비전 영상과 3차원 영상간의 정합 방법에 대하여 기술한다. 4장에서는 로봇의 경로구현을 위한 알고리즘을 제안하고, 5장에서는 구현된 시스템의 시뮬레이션 및 실험 결과를 비교 제시함으로 제안된 시스템의 유용성을 보이고, 마지막 6장에서 결론을 제시한다.

II. 이동 로봇의 위치 판단을 위한 localization과

Landmark 인식을 위한 패턴 인식

이동 로봇의 정확한 위치를 계산하기 위해서 적용된 localization 방법은 linear position estimation 알고리즘[4]을 사용함으로, 비선형 방정식들의 최소제곱해를 이용하는 Triangulation법[1,2]보다 우수한 연산 속도를 갖도록 함으로 이동 로봇의 정확한 위치를 빠른 시간 내에 계산할 수 있도록 하였다. 이 방법은 로봇이 인식하는 Landmark들을 복소평면 상의 점으로 간주하여 선형 벡터 방정식의 집합으로부터 해를 유도할 수 있다.

또한 로봇 측위를 수행하기 위해서는 카메라 영상을 통해 입력된 Landmark를 정확하게 인식을 위한 영상처리 및 신경회로망을 이용한 패턴 인식 방법을 제안한다.

1. 로봇 위치 계산을 위한 linear estimation algorithm [4]

로봇 중심 좌표계에서 Landmark z 를 복소수로 표현하면, 각각의 Landmark $z_i^{(r)}$ 는

$$z_i^{(r)} = l_i e^{j\tau_i} \quad \text{for } i=1, \dots, n \quad (1)$$

와 같이 표현할 수 있다. 여기에서, 길이 l_i 은 로봇으로부터 Landmark $z_i^{(r)}$ 까지 알려지지 거리이며, 각도 τ_i 는 $z_i^{(r)}$ 와 좌표축 $x^{(r)}$ 사이의 측정된 각이다.

가장 왼쪽의 Landmark $z_0^{(r)}$ 을 참조 Landmark로 설정하고,

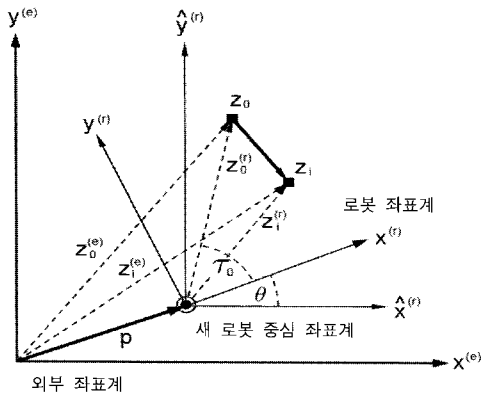


그림 1. 이동 로봇 좌표계.
Fig. 1. Coordinate of mobile robot.

Landmark z_0 와 Landmark z_i 사이의 시각(視角, visual angle) φ_i 를 다음과 같이 계산한다.

$$\varphi_i = \tau_i - \tau_0 \quad \text{for } i=1, \dots, n \quad (2)$$

각각의 벡터 $z_1^{(r)}, \dots, z_n^{(r)}$ 를 $z_0^{(r)}$ 로 나누고 $z_i^{(r)} = z_0^{(r)} + v_i^{(r)}$ 를 대입하여 다시 정리하면, 벡터 $z_0^{(r)}$ 과 $v_i^{(r)}$ 그리고 비 l_i/l_0 로 표현되는 다음과 같은 식을 얻을 수 있다.

$$\frac{1}{z_0^{(r)}} = \frac{l_i}{l_0} \frac{1}{v_i^{(r)}} e^{j\varphi_i} - \frac{1}{v_i^{(r)}} \quad (i=1, \dots, n) \quad (3)$$

$z_0^{(r)}$ 을 소거하기 위해, (3)의 왼쪽 항을 오른쪽 항의 k 인덱스 식으로 대체하면,

$$\frac{l_k}{l_0} \frac{1}{v_k^{(r)}} e^{j\varphi_k} - \frac{1}{v_k^{(r)}} = \frac{l_i}{l_0} \frac{1}{v_i^{(r)}} e^{j\varphi_i} - \frac{1}{v_i^{(r)}} \quad (i=1, \dots, n), k \neq i \quad (4)$$

과 같은 식을 얻을 수 있다. (4)에서 벡터 $v_i^{(r)}$ 와 l_i/l_0 만 알려지지 않은 인수이고, 각 $\varphi_1, \dots, \varphi_n$ 은 로봇중심 좌표계의 방향에 대해 독립적이기 때문에, (4)은 다른 방향을 갖는 로봇중심 좌표계에 대해 다음과 같이 표현될 수 있다.

$$r_k \frac{1}{u_k^{(r)}} e^{j\varphi_k} - \frac{1}{u_k^{(r)}} = r_i \frac{1}{u_i^{(r)}} e^{j\varphi_i} - \frac{1}{u_i^{(r)}} \quad (i=1, \dots, n), k \neq i \quad (5)$$

여기에서, $r_i = |z_i^{(r)}|/|z_0^{(r)}|$ 과 $u_i^{(r)} = z_i^{(r)} - z_0^{(r)}$ 만약, 로봇 중심 좌표계를 외부좌표계와 같은 방향으로 설정한다면 로봇 중심 좌표계의 $x^{(r)}$ 와 $y^{(r)}$ 좌표축은 외부 좌표계의 $x^{(e)}$ 와 $y^{(e)}$ 에 각각 평행하게 되며,

$$u_i^{(r)} = v_i^{(e)} = z_i^{(e)} - z_0^{(e)} \quad (6)$$

을 만족하게 된다. 따라서 (5)와 (6)으로부터 다음 식을 유도할 수 있다.

$$r_k \frac{1}{v_k^{(e)}} e^{j\varphi_k} - \frac{1}{v_k^{(e)}} = r_i \frac{1}{v_i^{(e)}} e^{j\varphi_i} - \frac{1}{v_i^{(e)}} \quad (i=1, \dots, n), k \neq i \quad (7)$$

(7)에서 알려지지 않은 인수는 오직 비 r_1, \dots, r_n 만 남게 된다. 여기서, $x^{(r)}$ 와 $y^{(r)}$ 축의 로봇 중심좌표계를 이용하는 것이 로봇 자신이 향하고 있는 방향 θ 를 이미 알고 있다는 것을 의미하는 것은 아니다. 하지만, (7)로부터 일단 $z_0^{(r)}$ 을 계산해 내면 다음 식으로부터 로봇이 향하고 있는 방향 θ 를 계산해 낼 수 있다.

$$\theta = \angle(z_0^{(r)}, x^{(r)}) - \tau_0 \quad (8)$$

(7)는 $Ar=c$ 형태의 행렬 형태로 변환될 수 있다. 여기에서 행렬 A 는 복소수 $b_i = (b_{\varphi_i}, b_{y_i}) = c_i e^{j\varphi_i}$ 로 구성된 $n(n-1)$

1) × n 행렬이며, 벡터 $r = (r_1, \dots, r_n)$ 은 벡터 $z_1^{(r)}, \dots, z_n^{(r)}$ 의 아직 알려지지 않은 길이의 비이다. c 는 복소수 $c_i = (c_{\xi}, c_{\eta}) = 1/u_i^{(r)}$ 의 차로 구성된 $n(n-1)$ 차의 벡터이다. A 행렬과 c 벡터를 풀어서 정리하면 다음과 같다.

$$A = \begin{pmatrix} b_1 - b_2 & 0 & 0 & \dots & 0 & 0 \\ b_1 & 0 & -b_3 & 0 & \dots & 0 & 0 \\ b_1 & 0 & 0 & -b_4 & \dots & 0 & 0 \\ \vdots & & & & & & \\ b_1 & 0 & 0 & 0 & \dots & 0 & -b_n \\ -b_1 & b_2 & 0 & 0 & \dots & 0 & 0 \\ 0 & b_2 & -b_3 & 0 & \dots & 0 & 0 \\ 0 & b_2 & 0 & -b_4 & \dots & 0 & 0 \\ \vdots & & & & & & \\ 0 & b_2 & 0 & 0 & \dots & 0 & -b_n \\ \vdots & & & & & & \\ -b_1 & 0 & 0 & 0 & \dots & 0 & b_n \\ 0 & -b_2 & 0 & 0 & \dots & 0 & b_n \\ 0 & 0 & -b_3 & 0 & \dots & 0 & b_n \\ \vdots & & & & & & \\ 0 & 0 & 0 & 0 & \dots & -b_{n-1} & b_n \end{pmatrix} \quad (9)$$

$$c = \begin{pmatrix} c_1 - c_2 \\ c_1 - c_3 \\ c_1 - c_4 \\ \vdots \\ c_1 - c_n \\ c_2 - c_1 \\ c_2 - c_3 \\ c_2 - c_4 \\ \vdots \\ c_2 - c_n \\ \vdots \\ c_n - c_1 \\ c_n - c_2 \\ c_n - c_3 \\ \vdots \\ c_n - c_{n-1} \end{pmatrix} \quad (10)$$

선형방정식 계인 $Ar=c$ 은 n 개의 미지수(r_1, \dots, r_n)에 $n(n-1)$ 개의 방정식을 갖는 과대결정계이다. 그러므로 측정 에러에 의해 불능이 될 수 있다. 그러므로 $n(n-1)$ 방정식의 평균 에러를 최소화하기 위해 최소제곱법을 이용하여 r 을 구한다.

평균 에러를 다음과 같이 정의한다.

$$E^2 = \|Ar - c\|^2 \quad (11)$$

만약, 정확한 해가 있다면 에러 E 는 0이다. 그러나 실제로 에러가 없는 상황은 존재할 수 없다. 그러므로 E 를 최소로 하는 문제로 접근해야 한다. E^2 이 최소가 되기 위한 필수 조건은 이것의 벡터 r 에 대한 미분이 0이 되는 점이다.

$$\begin{aligned} \frac{d}{dr} E^2 &= \frac{d}{dr} \|Ar - c\|^2 \\ &= \frac{d}{dr} (r^T A^T A r - r^T A^T c - c^T A r + c^T c) \\ &= 2A^T A r - 2A^T c \end{aligned} \quad (12)$$

즉,

$$0 = 2A^T A r - 2A^T c \quad (13)$$

$$r = (A^T A)^{-1} A^T c \quad (14)$$

여기서, $A^T A$ 는 nonsingular 행렬이다. (14)로부터 r_i 인자를 갖는 벡터 r 을 계산해 내고, 다음 식을 통해 어떤 임의의 i 에서 $z_0^{(r)}$ 를 계산해 낼 수 있다.

$$\frac{1}{z_0^{(r)}} = r_i \frac{1}{v_i^{(e)}} e^{j\phi_i} - \frac{1}{v_i^{(e)}} \quad (15)$$

(15)는 (3)으로부터 유도할 수 있으며, 이 식으로부터 일단 $z_0^{(r)}$ 이 계산되면 로봇의 위치 p 는

$$p = z_0^{(e)} - z_0^{(r)} \quad (16)$$

로부터 계산할 수 있다.

2. 신경회로망 Landmark 패턴 인식

비전 시스템에서 캡처된 영상으로부터 인식 가능한 Landmark를 추출해 내고 신경회로망을 이용하여 인식하는 이미지 프로세싱 절차는 다음과 같다.

2.1 Landmark 영상 처리

로봇의 위치 인식을 위해 사용된 식별 마크(landmark)의 형태는 그림 3과 같이 제안하였다. 비전을 통해 얻어진 영상을 이진화 과정을 거쳐 관심영역만을 추출한 후, 노이즈를 제거하고 중앙 마크를 분리하는 과정을 거친다.

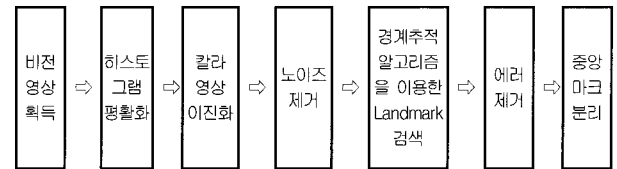


그림 2. 영상처리 절차.

Fig. 2. Procedure of image processing.

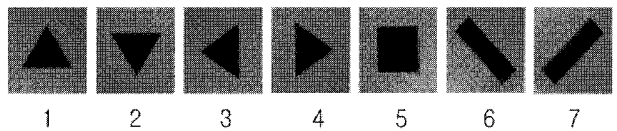


그림 3. 사용된 landmark.

Fig. 3. Used landmark.

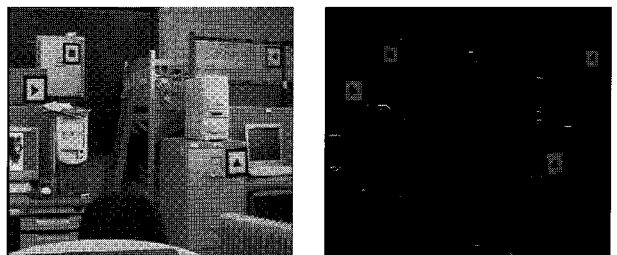


그림 4. 이진화 변환 전영상과 후영상.

Fig. 4. Binary image before and after conversion.

2.2 Landmark 인식을 위한 신경망 패턴인식

추출된 Landmark는 신경회로망 패턴 인식을 위하여 분리된 마크로부터 feature 값들을 찾아내고, 하나의 마크 당 총 17개의 feature가 정의되며 마크의 회전은 고려하지 않았다. 17개의 값 중에서 처음 8개를 찾기 위해 경계추적 알고리즘을 적용하고, 이것들로 모든 마크를 구별하기 힘들어 나머지 9개의 특징을 나타내는 값들을 활용한다. 따라서 제시된 마크의 특징 값은 다음과 같이 정의된다.

$$feature = [b, a + c, 0, d + f, e, g, F_9, F_{10}, F_{11}, \dots, F_{16}, F_{17}] \quad (17)$$

2.3 신경회로망

패턴인식을 위해 다층퍼셉트론 구조는 다층퍼셉트론으로 구성했으며, 학습과 예측을 위해서 역전파(back-propagation) 알고리즘을 적용하였다[5]. 신경회로망 구조는 입력 층은 18개의 뉴런으로 구성되었으며, 은닉 층은 6개, 출력 층은 1개의 뉴런으로 구성된다. 활성화 함수는 은닉 층, 출력 층 모두 다음 식을 사용하였고

$$\Gamma(x) = \frac{1}{1 + \exp(x)} \quad (18)$$

표 1. 특징 정의 예.

Table 1. The example of defining features.

	번호	픽셀 수
	1(→)	b
	2(↘)	a+c
	3(↓)	0
	4(↙)	0
	5(←)	d+f
	6(↗)	0
	7(↑)	e
	8(↖)	g

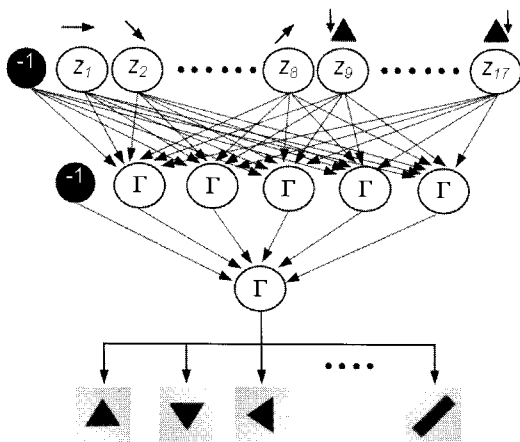


그림 5. 패턴 인식을 위한 신경망 구조.

Fig. 5. Neural networks structure for pattern recognition.

여기에서 Bias는 -1로 설정하고, 앞에서 구한 17개의 feature 값이 입력된다.

III. 로봇 측위 시스템과 3차원 영상과의 Overlay

앞 장에서 제시한 로봇 위치 인식 알고리즘에 의해 계산된 위치와 비전 영상과 3차원 VRML 영상의 정합에 의해 정확한 Localization 정보를 얻음으로써 보다 정확한 로봇의 경로를 구현하기 위해 제안한 방법은 그림 6과 같다.

1. 비전을 통한 이동 로봇의 Landmark 식별 과정

① 비전으로부터 Landmark가 분포해 있는 작업 환경 영상을 획득한다.

② 제안한 영상처리 기법들을 통하여 비전 영상으로부터 Landmark를 추출해 낸다.

③ 다음으로 신경회로망 패턴인식 기법을 통하여 Landmark를 식별하고, Landmark들 간의 시각(φ)과 전체좌표상의 위치(z)를 찾는다. 시각(φ)는 다음과 같이 계산한다.

그림 7과 같이 Landmark z_0 와 z_i 사이 각을 ϕ_i 라 하면 $\varphi_i = \alpha_0 + \alpha_i$ 이다. 중심선으로 부터 각 Landmark 까지 각을 나타내는 각 α_i 는

$$\alpha_i = \tan^{-1} \left(\frac{2p_i}{p_h} \tan(\delta/2) \right) \quad (19)$$

여기서, p_h 는 비전 영상의 가로 픽셀 수이며, δ 는 비전의 화각이다. 본 연구에서는 $p_h = 320$ 으로 $\delta = 33^\circ$ 로 설정하였다.

④ 시각(φ)과 전체좌표상의 위치(z)을 이용하여 2장에서 제시했던 linear position estimation 알고리즘에 의해 로봇의 위치(x, y)와 진행방향(θ)를 계산해 낸다. 여기서 계산된 x, y, θ 는 비전영상의 왜곡이나 잘못된 Landmark식별에 의해 에러를 포함할 수 있다.

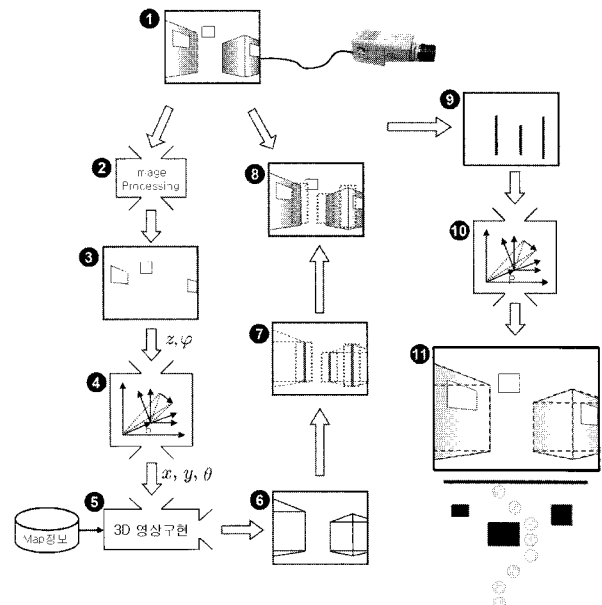


그림 6. 제안한 시스템의 path finding 방법.

Fig. 6. Path finding method of proposed method.

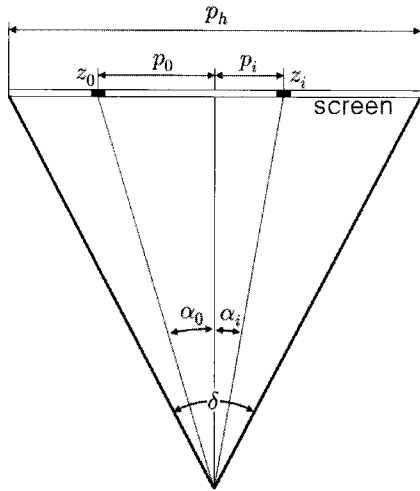


그림 7. Landmark 간의 시각(視角) 측정.
Fig. 7. Measurement of visual angle between landmark.

⑤ ④에서 계산한 로봇의 x, y, θ 와 이미 작업환경의 정보(Map 정보)를 담고 있는 VRML데이터를 바탕으로 3차원 영상을 구현한다.

⑥ ⑤에서 구현한 3차원 영상은 wire-frame으로 구현된다.

⑦ 3차원 영상으로부터 세로 직선들만 추출한다. 그리고 추출한 직선들의 주위로 ⑦의 그림과 같이 일정 범위(점선 영역)를 설정하고 이를 ‘매칭 가영역’이라 부르기로 한다.

⑧ ⑦에서 설정한 ‘매칭 가영역’을 비전 영상에 적용한다. 매칭 가영역은 3차원 영상 내의 세로 직선을 l_{3D} 라 하고 비전영상 내의 세로 직선을 l_{vision} 이라 하면 l_{3D} 와 일치하는 l_{vision} 이 존재 가능한 영역을 의미한다.

⑨ 비전 영상으로부터 매칭 가영역내의 세로 직선을 추출해 낸다. 세로 직선의 추출 방법은 다음 절에서 설명한다. 만일 이 직선들의 위치와 3차원 영상 직선의 위치가 서로 일치한다면, ④에서 계산한 로봇의 x, y, θ 는 정확하다고 판단할 수 있다. 즉, 두 영상의 직선들이 일치했을 경우는 더 이상 진행할 필요 없이 로봇의 위치를 x, y, θ 로 결정한다.

⑩ 그러나 두 영상의 직선들이 일치 하지 않았을 경우에는 ④에서 계산한 x, y, θ 에 에러를 포함하고 있음을 알 수 있다. 하지만 ⑨에서 추출한 세로 직선의 위치와 그 세로 직선과 일치하는 3차원 영상내의 직선의 데이터(전체 좌표상의 위치)를 알기 때문에 이를 이용하여 다시 한 번 2장의 localization을 수행할 수 있다. 결국, ④과정에서 에러를 포함하는 로봇의 위치를 3차원 영상을 이용하여 유효성을 검증하고, 에러가 클 경우에는 보다 정확한 Landmark를 이용하여 localization을 수행하는 것이다.

⑪ ⑩여기에서 다시 계산한 로봇의 위치 x, y, θ 에 의해 3차원 영상을 재구성하고, 3차원영상을 비전영상 위에 Overlay를 한다. 획득된 로봇의 x, y, θ 를 바탕으로 경로를 찾고 로봇의 길은 2차원 공간에서 나타내어지게 된다.

그림 6의 ②-④의 localization 절차는 다른 시스템으로

대체 될 수 있다. 예를 들어 초음파나, 관성항법장치, 엔코더 등을 이용한 위치 추정 알고리즘에 의해 로봇 위치를 계산하고 이들 계산 값들이 정확한지 ⑤-⑨과정을 통해 검증하고 에러가 있을 경우, 나머지 과정을 통해 보다 정확한 localization을 수행하는 할 수 있다. 즉, 다른 장치들과 비전과 3차원 맵에 의한 하이브리드형 측위 시스템이 구현 가능하다.

2. 영상 내의 세로선 추출 방법

영상 내에서 직선을 검출하기 위해서는 영상의 에지 추출과 에지 영상으로부터 직선의 방정식을 찾는 절차를 수행해야 한다. 영상 에지 추출을 위해 다음과 같은 prewitt 마스크를 사용한다.

주로 X방향과 Y방향으로의 밝기 값의 차이를 구하기 위해 마스크를 적용하며 결과는 두 방향의 크기를 이용하여 구하게 된다.

영상에서 직선을 방정식을 추출하는 알고리즘 중 가장 널리 쓰이는 방법은 Hough transform이다. 영상에서 직선을 추출하는 예를 살펴보면 다음과 같다. 일반적인 형태의 직선의 방정식 $y = ax + b$ 형식은 기울기가 무한대가 될 수 있기 때문에 다음의 직선 방정식을 사용한다.

$$\rho = x \cos \theta + y \sin \theta \tag{20}$$

그림 9와 같이 ρ 는 원점에서 직선까지의 법선거리를 나타내며, θ 는 각도를 나타낸다. (x, y) 평면에서 표시되는 각각의 1, 2, 3은 (ρ, θ) 평면에서 오른쪽 그림과 같이 곡선으로 표시된다. 이평면에서 세 곡선의 교차점이 ρ 와 θ 를 결정짓게 되며 이 인수로 인하여 (20)으로부터 직선의 방정식을 구할 수 있게 된다.

이 연구에서는 $\theta = 0$ 인 세로축에 평행한 직선만을 찾기 때문에 (20)은 $\rho = x$ 가 되며 결국 가장 많은 픽셀을 포함한 ρ 값에서 직선의 방정식을 찾게 된다.

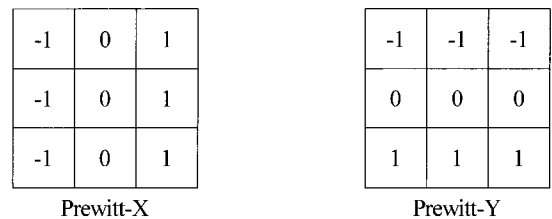


그림 8. Prewitt 마스크.
Fig. 8. Prewitt mask.

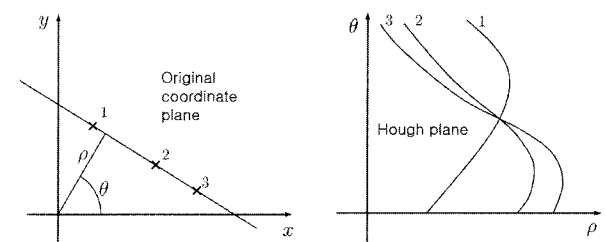


그림 9. 일반 xy 좌표계와 Hough 좌표계.
Fig. 9. General x-y and Hough coordinate system.

IV. 자율주행 이동 로봇의 전역 경로 계획

우리의 검색지역은 사각형 격자들로(Grid) 나누어져 있다. 격자들로 지역을 단순화함으로 검색지역을 단순한 2차원 배열로 만들어 준다. 이는 3차원의 VRML의 좌표를 단순한 2차원 배열로 만들어주고, 각각의 아이템(시작점, 도착지점, 길 등)은 사각형 격자위에 하나의 2차원 배열로 묘사된다. 사각형의 길은 사각형의 중앙에서 다음 사각형의 중앙으로 이동하며 목표에 도착할 때까지 반복 수행한다 [17].

길을 찾을 때 사각형을 선택하는 방법은 다음방정식을 따른다.

$$F = G + H \quad (21)$$

G는 출발점 A로부터 얻어진 새로운 사각형까지의 이동 비용이다. 길을 찾아가면서 생성되게 된다. H는 얻어진 사각형으로부터 최종 목적지점까지의 예상비용이다. 이 값은 매번 조회될 때마다 발견된다. 이 값은 매번 조회될 때마다 발견되며, 추측 값이기 때문에 완전한 길을 찾거나 정확한 거리를 알 수 없다. 이는 모든 장애물이나 벽 등이 길 위에 있기 때문이다.

길은 열린 목록과 사각형의 작은 F값 선택을 반복함으로써 얻어진다. 먼저 사각형 이동에 있어서 가로 또는 세로로 이동하는데 각각 10의 비용을 할당한다. 그리고 대각선 이동에 대해서는 14의 비용을 할당한다. 대각선으로 이동하는 비용은 $\sqrt{2}$, 즉 1.414로 많은 계산 량을 피하기 위해서 14의 비용을 할당하게 된다. 그러므로 루트계산과 10진법을 피하게 되는 것이다. G비용의 계산은 부모로부터 G비용을 가져와서, 그 부모로부터 직각으로의 이동은 10을 대각선의 이동은 14를 할당한다. H는 길들의 변화 안에서 어렵잡아 예측하게 되는데, 대각선이동을 제외한 가로 또는 세로 이동한 총 숫자를 계산 하게 된다. H비용을 계산할 때는 어떤 끼어드는 방해물도 무시한다. 실 거리가 아닌 어렵잡은 추측이기 때문이다.

마지막으로 G와 H를 더함으로써 F가 계산된다. F, G, H는 각각의 사각형 안에 표시된다. 그림 5.4에서 보는바와

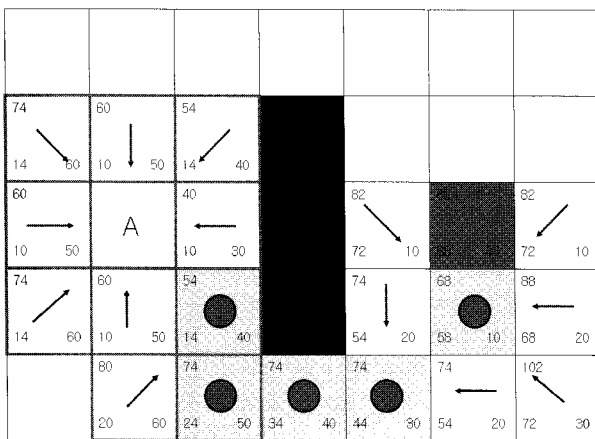


그림 10. 최종 목표점까지의 경로 검색.
Fig. 10. Path finding to the final goal.

같이 F는 좌상단에 G는 좌하단 그리고 H는 우하단에 표시 되어있다. 시작점으로부터 위아래 옆의 G비용은 10이다. 대각선으로 위치한 사각형들은 14의 G비용을 갖는다. 각각의 H비용은 장애물을 무시한 시작점으로부터 목표점까지의 가로 세로방향으로만 사각형의 수를 계산한다. 즉 시작사각형의 오른쪽에 있는 사각형은 목표사각형으로부터 3개의 사각형만큼의 거리에 있으므로 30의 비용을 할당받게 되고, 이 사각형 위에 있는 사각형은 4개의 사각형만큼 떨어져 있으므로 40의 비용을 할당받게 되는 것이다.

V. SIMULATION 및 결과 고찰

1. 실험 환경 구축

시뮬레이션을 통한 제안된 알고리즘의 유용성을 보이기 위해 구성된 실험 장비 사양은 다음과 같다. 실험에 사용된 로봇 작업 환경의 크기는 600×800mm²의 모형을 사용하였다.

2. 설정된 좌표계

실제 모델에서의 좌표계는 그림 11과 같이 설정한다.

표 2. 실험 장비 사양.

Table2. Specification of the experimental equipment.

장 비	특 징
카메라 본체	기종 : SDC-411 촬영소자 : 1/3" CCD 41만 화소 촬영면적 : 4.88(H) × 3.66(V)mm ²
카메라 렌즈	기종 : SLA084C 초점거리 : 8mm 최대구경비 : 1:1.4 후초점거리 : 14.137mm
캡처보드	DVD PLUS 캡처영상 크기 : 320(H)×240(V) pixels
컴퓨터	CPU : Pentium IV 2.4G RAM : 512MB OS : Windows XP SP2 프로그래밍언어 : Visual C++ 6.0

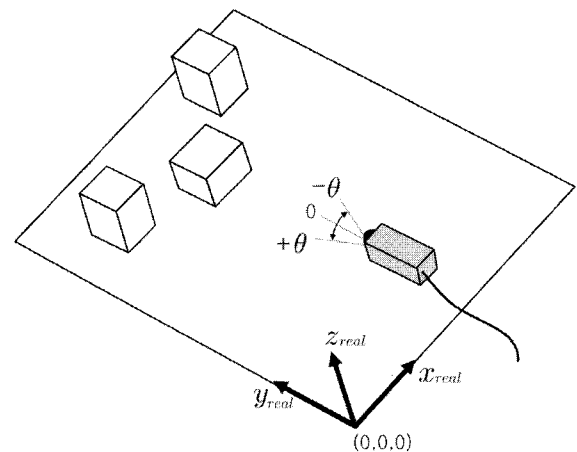


그림 11. 실 모델 좌표계 설정.

Fig. 11. Definition of coordinate of real model.

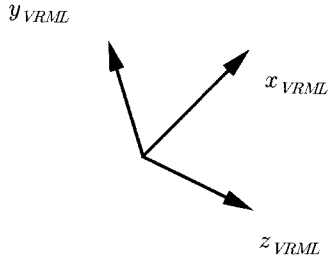


그림 12. VRML 좌표계.
Fig. 12. Coordinate of VRML.

모형 바닥면의 왼쪽 아래 모서리를 원점으로 설정하고 좌표축을 그림처럼 설정한다. 카메라의 각도는 왼쪽을 (+) 방향, 오른쪽을 (-)방향으로 설정한다. VRML에서의 좌표계는 실 모델 좌표계와 약간 다르다. VRML의 좌표계는 기본적으로 그림 12와 같이 설정되어 있다.

그러므로 localization 결과의 x, y 좌표는 VRML에 적용될 때 다음 식에 따라 적용되어야 한다.

$$\begin{aligned} x_{VRML} &= x_{real} \\ y_{VRML} &= z_{real} \\ z_{VRML} &= -y_{real} \end{aligned} \quad (22)$$

3. 실험 결과 및 고찰

3.1 Feature 추출 및 패턴인식을 측정 실험 방법

Feature 데이터 추출 및 패턴 인식을 측정 실험 장비를 그림 13과 같이 세팅한다.

Landmark 하나를 벽면에 붙이고 카메라로 직접 영상을 입력받아 영상 처리를 통해 feature 값을 추출한다. 카메라의 위치를 수시로 옮겨가며 영상을 입력받으며 카메라의 각도는 Landmark가 영상에 들어올 수 있는 범위 내에서 무작위로 설정한다. 각 Landmark당 50개의 feature 데이터를 추출하여 신경망을 학습한다.

신경회로망 학습 후 역시 같은 방법으로 학습에 대한 검증 실험을 실시한다. feature 데이터 추출과 마찬가지로 다양한 각도와 위치에서 보여 지는 Landmark의 영상을 인식한다.

3.2 신경회로망 패턴인식을 실험 결과

신경회로망은 0.1에서 0.7사이 값을 출력하며, 출력을 소수점 2째 자리에서 반올림하고 10을 곱한 결과의 정수부분을 인식한 패턴 번호로 한다. 표 3은 인식 결과를 정리한 것이다. 인식 오류는 대부분 카메라와 Landmark가 멀리 떨어져 있거나 주위 빛의 양이 적어 Landmark 추출이 용이하지 않을 때 발생하였다. 주위 환경의 빛의 양을 일정하게만 해 주어도 인식률은 크게 상승하였다.

3.3 Landmark-localization 실험 방법

실제 작업환경에서는 로봇이 인식할 수 있는 Landmark들을 동서남북 4방향 모두에 설치할 해야 하지만 이번 실험에서는 한 쪽 방향에만 Landmark를 설치하고 한 방향에 대한 실험만 실시하였다.

실험 모형의 바닥에는 실제 카메라의 위치를 측정할 수 있도록 눈금이 매겨져 있고, 카메라에는 카메라가 향하는 각도를 측정할 수 있도록 각도기를 설치하였다. 카메라를

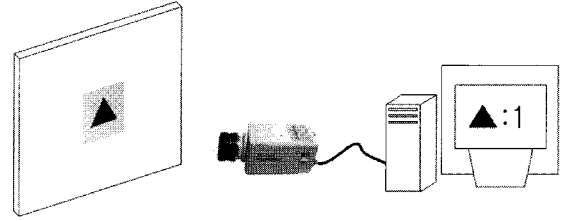


그림 13. 패턴인식 측정 실험 방법.
Fig. 13. Method of pattern recognition in experimentation.

표 3. 신경회로망 패턴 인식률.

Table 3. The rate of pattern recognition by using neural network.

패턴번호	Landmark	인식률
1		97.8%
2		99.0%
3		98.0%
4		98.8%
5		97.9%
6		98.0%
7		99.4%

눈금위에 무작위로 위치시키고 Landmark를 향한 후 localization을 수행한다. Localization 결과와 실제 카메라의 위치를 눈금으로부터 읽어 비교하고 정확도를 계산한다. 실험은 위치 측정과 방향각 측정 두 가지로 나누어 실시하였다. 위치 측정은 카메라의 각도를 0°로 고정하고 다양한 위치에서 수행하였으며, 방향각은 임의의 위치에서 -50°에서 50°까지 1°간격으로 카메라의 방향을 전환하며 측정하였다.

3.4 실험 결과 및 고찰

위치 보정 실험을 위해 Landmark에 노이즈를 더하여 일부러 측위 결과에 에러를 갖도록 하였다. 그리고 VRML 3D 영상을 이용한 위치예러 보정을 실시하였다.

VRML을 이용한 위치 예러 보정 결과는 그림 14와 같다. 보정전의 위치예러는 x, y 좌표 각각 -14mm에서 +15mm까지 넓은 분포를 보이지만 보정 후의 x, y 좌표는 거의 대부분의 예러 값이 -4mm에서 +4mm 사이에 분포한다. 보정 전의 위치예러가 x 방향으로 편향 되어 있는 이유는 Landmark가 보다 (-)방향에 있는 것처럼 인식 하도록 노이즈 추가했기 때문에 나타난 결과이다.

VRML을 이용한 방향 각 예러 보정 결과는 다음과 같다. 보정 전의 방향 각 예러는 -0.6°에서 +1.6°의 분포로 대부분의 예러들이 큰 값을 갖지만, 보정 후의 방향 각 예러는 -0.6°에서 +0.6°의 사이에 분포하고 큰 예러 값보다 작은 예러 값들의 빈도수가 더 높게 나타났다.

표 4에서 보는바와 같이 VRML을 이용한 위치 오차 보

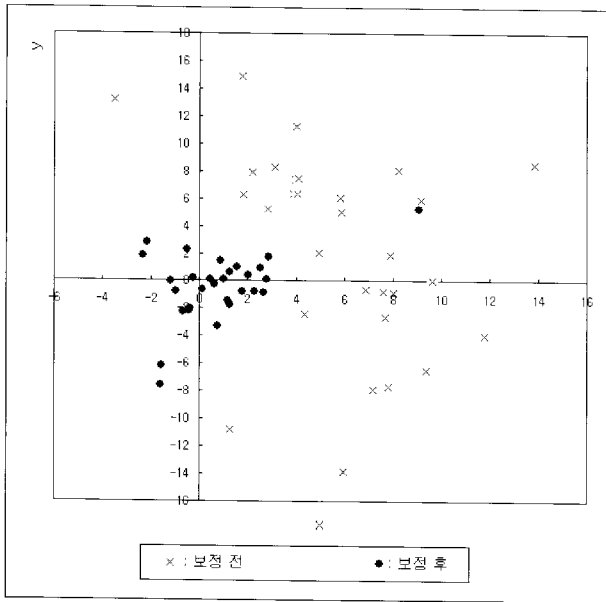


그림 14. VRML이용한 에러 보정 결과(위치).
 Fig. 14. The result of error compensation by using VRML (position).

표 4. VRML을 이용한 오차 보정 후 데이터 통계.

Table 4. Data statistics after error compensation by using VRML.

	보정 전			보정 후		
	x	y	θ	x	y	θ
평균	5.719	2.464	0.641	0.751	-0.479	0.034
표준편차	3.443	7.0	0.455	2.127	2.438	0.271
				-38.2%	-65.2%	-40.4%

표 5. Landmark-Localization과 VRML을 이용한 보정 Localization 결과 비교.

Table 5. The result of comparison by using landmark-localization and VRML compensation.

	Landmark-Localization			VRML을 이용한 보정		
	x	y	θ	x	y	θ
평균	0.758	0.261	0.162	0.751	-0.479	0.034
표준편차	1.419	2.955	0.317	2.127	2.438	0.271
				+49%	-17.5%	-14.5%

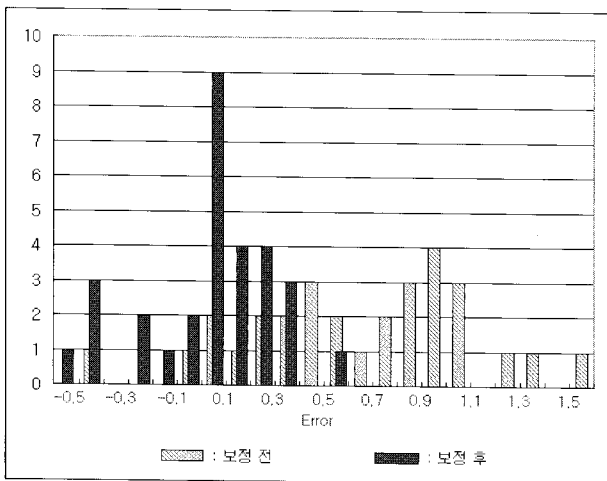


그림 15. VRML이용한 에러 보정 결과(방향 각).
 Fig. 15. The result of error compensation by using VRML(each direction).

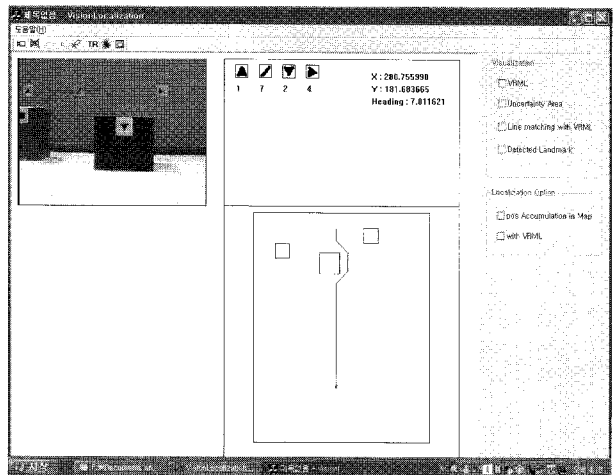


그림 16. 최종 시뮬레이션 결과.
 Fig. 16. The final result of simulation.

정 후, x 좌표는 38.2%, y 좌표는 65.2%, θ 는 40.4%의 오차 감소를 보였다.

Landmark-localization의 결과와 VRML을 이용한 오차보정의 결과는 표 5와 같다. x 좌표에 대한 오차는 49%정도 컸으나 y 좌표의 오차는 17.5%, θ 의 오차는 14.5%의 감소를 보였다. x 좌표의 경우 49%의 오차 증가를 보였으나 실제 수치로는 0.7mm 정도로 큰 차이를 보이지 않았다.

이 Landmark-localization과 VRML을 적용하여 위치 오차를 최소한으로 감소할 수 있었고, 최적 경로 계획을 위하여 A*알고리즘을 적용한 시뮬레이션 결과는 그림 16과 같다. 본 결과는 이동 로봇이 장애물을 회피하여 최적 경로를 잘 찾아 가고 있음을 보여 준다.

VI. 결론 및 향후 과제

본 논문에서는 로봇의 정확한 위치를 바탕으로 최적의 경로 설계를 위해 Landmark를 이용한 측위와 측위에서 발생하는 에러를 VRML 3D 영상과의 매칭에 의해 보정하는 시스템을 제안하였고, 최종적으로 이동 로봇의 경로 알고리즘을 구현하였다.

위와 같은 시스템을 구현하기 위해 영상 처리 기법, 신경망 패턴인식 기법, 그리고 효율적인 Localization 알고리즘과 3차원 영상구현을 위한 컴퓨터 그래픽스 기법, 그리고 이동 로봇 경로 계획을 위한 A*알고리즘 등이 적용되었다.

구현된 시스템의 수행과정은 다음과 같다. 먼저, 비전으로 입력받은 영상으로부터 작업 환경 주의에 산개 되어 있는 Landmark들을 영상 처리 기법을 통해 추출하였고, 이를

신경망 패턴인식 기법을 통해 인식을 한다. 인식된 Landmark의 정보와 Landmark 사이의 시각을 이용해 측위를 수행하였고, 이때 계산된 로봇의 위치와 방향각을 토대로 3차원 영상을 구현하였다. 3차원 영상과 비전영상의 정합정도를 파악하고 두 영상이 정합이 되지 않으면 3차원 영상의 세로 선과 일치하는 비전 영상내의 세로선을 찾는다. 찾아진 선을 Landmark로 하여 다시 측위를 수행하고 비전영상에 3차원 영상을 오버레이 함으로써 보다 정확한 로봇의 위치를 얻게 된다. 정확한 로봇의 위치는 로봇의 path 구현에 적용되고 이는 보다 정확한 경로를 구현할 수 있게 되었다.

제안된 방법에 의해 노이즈를 포함한 위치와 방향각 데이터의 에러를 약 30~60% 감소시켰으며, 노이즈를 포함하지 않은 Landmark-localization과의 비교에서도 거의 동등한 성능을 나타내는 결과를 얻었다.

비전 영상에 3차원 영상을 오버레이 함으로 얻어진 장점은 두 영상의 오버레이를 통해 두 영상의 정확한 매칭을 통한 측위 결과의 정확성을 검증할 수 있었다. 또한 VRML에 의한 3차원 영상은 작업 환경의 정보를 담고 있기 때문에, 레퍼런스로서 활용됨으로 장애물의 뒷면이나 사물의 내부까지 파악이 가능함을 알 수 있었다.

본 연구의 향후 과제는 이동 로봇의 안전한 임무 수행을 위해 보다 안전한 곡률 경로 설계에 관한 연구가 필요하다.

참고문헌

[1] C. B. Madsen, C. S. Andersen, "Optimal landmark selection for triangulation of robot position," *Robotics and Autonomous Systems*, vol. 23, Issue 4, pp. 277-292, July 1998.

[2] K. Briechele and U. D. Hanebeck, Member, "Localization of a mobile robot using relative bearing measurements," *IEEE Transaction on Robotics and Automation*, vol. 20, no. 1, pp. 36-44 Feb 2004.

[3] E. Krotkov, "Mobile robot localization using a single image," *IEEE International Conference on Robotics and Automation*, vol. 2, pp. 978-983, May 1989.

[4] M. Betke and Leonid Gurvits, "Mobile robot localization using landmarks," *IEEE Transaction on Robotics and Automation*, vol. 13, no. 2, pp. 251-263 Apr. 1997.

[5] J. S. Esteves, A. Carvalho, C. Couto, "Generalized

geometric triangulation algorithm for mobile robot absolute self-localization," *Industrial Electronics, 2003. ISIE '03. 2003 IEEE International Symposium on*, vol 1, pp. 346-351, 9-11 June 2003.

[6] Cohen, Charles and Koss, V. Frank, "A comprehensive study of three object triangulation," *Mobile Robots VII*, SPIE vol. 1831, 1992.

[7] D. J. Kang and J. E. Ha, "Digital image processing using visual C++," *SciTech*, Korean, Mar 2003.

[8] <http://www.parallelgraphics.com/>

[9] J. Alex, B. Vikramaditya, and B. J. Nelson, "Tele-operated micromanipulation within a VRML environment using Java," *Intelligent Robots and Systems, 1998. Proceedings, 1998 IEEE/RSJ International Conference on*, vol. 3, pp. 1747-1752, 13-17 Oct. 1998.

[10] J.-Y. Huang, "Increasing the visualization realism by frame synchronization between the VRML browser and the panoramic image viewer," *International Journal of Human-Computer Studies*, vol. 55, Issue 3, pp. 311-336, Sep 2001.

[11] G. N. DeSouza and A. C. Kak, "Vision for mobile robot navigation: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 237-267, Feb. 2002.

[12] Wang, D, "Pattern recognition: Neural networks in perspective," *Expert, IEEE*, vol. 8, Issue 3, pp. 52-60, Aug. 1993.

[13] Antonios Gasteratos, Calos Beltran, Giorgio Metta, Giulio Sandini, "PRONTO: a system for mobile robot navigation via CAD-model guidance," *Microprocessors and Microsystems*, vol. 26, Issue 1, pp. 17-26, Feb 2002.

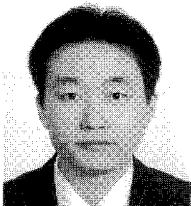
[14] <http://www.policyalmanac.org> A star path finding for Beginners

[15] R. SIEGWART, Illah R. NOURBAKHSH, "Introduce Autonomous mobile Robots"

[16] A. A. Razavian, J. Sun "Cognitive based adaptive path planning algorithm for autonomous robotic vehicles," *IEEE* 2005.

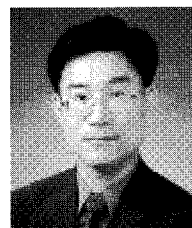
[17] <http://www.policyalmanac.org/>, "A* path finding for Beginners"

손 은 호

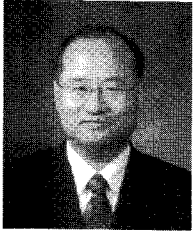


2005년 원광대 전지전자공학과 학사.
2006년 전북대 제어계측공학과 석사.
관심분야는 로봇 Localization, Robotics.

김 영 철



1989년 전북대학교 전기공학과 학사.
1988~1990년 LG 생산기술연구소 연구원.
1993년 전북대학교 전기공학과 석사.
1998년 일본 Tohoku대학 기계공학 박사.
1998년~2004년 대불대학교 전기전자공학부 전임강사. 조교수. 2005년~현재 군산대학교 기계공학부 부교수. 관심분야는 메카트로닉스, 재활복지공학, 로봇센싱, Robotics.



정길도

1984년 미국 Oregon State University
기계공학과 졸업. 1986년 미국 Georgia
Institute of Technology 기계공학석사.
1992년 미국 Texas A&M University 기
계공학박사. 1993년~1995년 영남대학
교 전임 강사. 1995년~현재 전북대학

교 전임강사, 부교수, 교수. 관심분야는 비선형 Time-Delay
시스템, Web 기술, Robotics.