

오차 감소를 위한 이동로봇 Self-Localization과 VRML 영상오버레이 기법

Self-localization of a Mobile Robot for Decreasing the Error and VRML Image Overlay

권 방 현, 손 은 호, 김 영 철, 정 길 도*

(Bang Hyun Kwon, Eun Ho Shon, Young Chul Kim, and Kil To Chong)

Abstract : Inaccurate localization exposes a robot to many dangerous conditions. It could make a robot be moved to wrong direction or damaged by collision with surrounding obstacles. There are numerous approaches to self-localization, and there are different modalities as well (vision, laser range finders, ultrasonic sonars). Since sensor information is generally uncertain and contains noise, there are many researches to reduce the noise. But, the correctness is limited because most researches are based on statistical approach. The goal of our research is to measure more exact robot location by matching between built VRML 3D model and real vision image. To determine the position of mobile robot, landmark-localization technique has been applied. Landmarks are any detectable structure in the physical environment. Some use vertical lines, others use specially designed markers, In this paper, specially designed markers are used as landmarks. Given known focal length and a single image of three landmarks it is possible to compute the angular separation between the lines of sight of the landmarks. The image-processing and neural network pattern matching techniques are employed to recognize landmarks placed in a robot working environment. After self-localization, the 2D scene of the vision is overlaid with the VRML scene.

Keywords : robot localization, VRML, landmark navigation

I. 서론

최근 이동 로봇은 다양한 분야에 사용되고 있고, 특히 원자로나 폭발물 처리 등과 같이 인간의 접근이 어려운 위험한 환경 내에서 그 역할은 증대되고 있다. 이동 로봇이 복잡한 작업 환경 내에서 정확하고 안전하게 원하는 작업을 수행하기 위해서는 자기 위치 파악 능력이 필요하다.

이동 로봇 위치 인식 기술은 크게 실내 환경과 실외 환경에서 자기 위치 파악 기술이 연구되고 있으며, 실외 환경의 경우 실내 환경에 비하여 많은 불확실성이 존재함으로 로봇 위치 인식률이 떨어진다. 실내에서 로봇의 자기 위치 인식은 실제 이미지와 목표 이미지 사이의 모델 매칭 기술을 활용한 맵 기반 이동(map-based navigation), 맵 형성 기반 이동(map-building-based navigation), 맵없는 이동(mapless navigation) 방법 등이 연구되어 지고 있다[1,2].

맵기반 위치 인식 기술은 로봇 이동 중에 CCD 카메라를 통해 입력되는 landmark의 정보와 실제 영상 정보와의 오차를 비교하여 위치 인식을 수행하나, 카메라와 landmark사이의 거리, 조명의 밝기 등에 의하여 위치 오차가 커지는 문제점과 landmark의 잘못된 인식이나 landmark 사이 각의 잘

못된 추정으로 인해 로봇 위치 계산결과는 항상 에러를 포함하고 있다[3]. 또한 위치 계산을 위해 제안된 기존의 triangulation 알고리즘은 비선형 방정식의 복잡한 폐쇄형 해(closed-form solution)로부터 로봇의 위치를 찾을 수 있으나 이와 같은 비선형 방정식을 다루는 계산 알고리즘은 실시간 이동로봇 위치 추적에 적용되기에는 처리 시간이 너무 오래 걸리는 문제점이 있다[2,4].

따라서 본 논문에서는 선형방정식의 최소제곱법을 적용한 linear position estimation 기법을 도입하여 기존의 제안된 triangulation 알고리즘의 계산 시간 문제를 해결하고, 최적의 landmark의 정보를 얻기 위하여 신경망 패턴인식 방법을 적용한다. 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 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 이동로봇 위치 인식 시스템의 전체 구성 방법을 설명하고, 3장에서는 신경망 패턴인식 방법을 적용한 landmark 추출 및 인식알고리즘을 제안한다. 4장에서는 이동 로봇의 자기 위치 계산을 위한 linear position estimation 알고리즘[3]을 도입하고, 5장에서는 계산된 localization 결과를 VRML에 적용하는 과정에 대해서 기술한다. 6장에서는 localization 시뮬레이션 및 실험 결과 비교를 통하여 제안된 시스템의 유용성을 보이고, 마지막 7장에서 결론을 제시한다.

* 책임저자(Corresponding Author)

논문접수 : 2005. 11. 25., 채택확정 : 2006. 2. 25.

권방현, 손은호 : 전북대학교 제어계측공학과

(iam720@chonbuk.ac.kr/yauchi1@chonbuk.ac.kr)

김영철 : 군산대학교 기계공학부(kimyc@kunsan.ac.kr)

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

※ 본 논문은 서울대 기초전력공학 공동연구소와 전북대 메카트로닉스 연구센터의 지원으로 연구를 수행함.

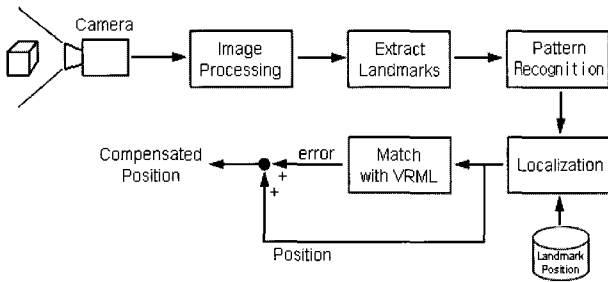


그림 1. 로봇 localization 시스템 다이어그램.
Fig. 1. A conceptual system diagram for robot localization.

II. 이동로봇 위치 인식 시스템의 구성

이동 로봇 위치 인식 시스템의 전체적인 구성은 그림 1 과 같다. CCD 카메라로부터 획득한 영상으로부터 다양한 영상처리 기법을 통해 landmark를 추출하고 신경망 패턴인식 기법을 통하여 인식된다. 이 landmark는 로봇의 자기 위치 결정에 사용되며, 이 때 사용하는 위치 인식 알고리즘은 기존에 많이 사용되었던 triangulation 알고리즘 대신 linear position estimation 기법[3]을 적용하여 구현하였다.

로봇 작업환경은 VRML 가상공간 모델로 미리 구축하고 localization 결과인 로봇의 위치와 방향각을 적용하여 가상 공간 상의 카메라에 의해 비쳐지는 3차원 영상을 얻는다. 이 가상영상과 CCD 카메라 영상과 비교를 통하여 Localization 결과를 검증하고 오차를 분석한다.

전체적인 시스템은 Visual C++ 6.0언어를 이용하여 구현하였고, Visual C++과 VRML의 연동을 위해 Java와 Java Script를 사용하였다.

III. 신경망 landmark 패턴 인식

카메라 영상을 통해 입력된 landmark의 식별을 위하여 신경망 패턴인식 알고리즘을 사용하였다.

1. Landmark 영상추출 과정

Landmark는 그림 2와 같이 채도가 높은 배경위에 식별 마크를 표시하였다. CCD 카메라로부터 얻어진 영상으로부터 landmark의 배경색 영역을 추출하고, 이 영역으로부터 landmark를 분리해 낸다. 이 과정에 적용된 영상처리 절차는 다음과 같다.

- 1) 컬러 히스토그램 스트래칭,
- 2) 컬러 영상 이진화,
- 3) Salt, Pepper 노이즈 제거,
- 4) 경계추적 알고리즘을 이용한 영역추출,
- 5) 원래 이미지로부터 landmark 분리,
- 6) 중앙의 마크만 추출, 이를 지정된 크기로 변환

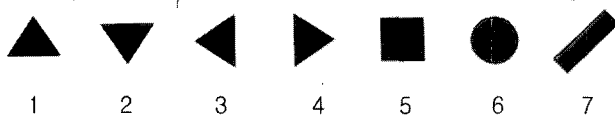


그림 2. 적용된 landmark.
Fig. 2. Landmarks used in the experiment.

2. Landmark 인식을 위한 신경망 패턴인식

2.1 패턴의 특징(feature) 정의

추출된 landmark는 신경회로망 패턴 인식을 위하여 분리된 마크로부터 feature 값들을 찾아낸다. 하나의 마크 당 총 17개의 feature가 정의되며 마크의 회전은 고려되지 않았다. 17개의 feature 값 중 처음 8개를 찾기 위해 경계추적 알고리즘을 사용한다.

현재 추적 중인 픽셀과 전 단계에서 추적된 픽셀의 위치 관계를 8-방향(→↘↓↙←↖↑↗)으로 정의하고, 이들 각 8 방향에 속하는 픽셀들의 수를 처음 8-특징으로 한다. 표 1에 포함된 이미지를 예로 들면, 첫 8-특징은 b, a+c, 0, 0, d+f, 0, e, g로 정의 된다. 여기서 알파벳 기호는 같은 방향으로 연속된 픽셀들의 개수를 나타낸다.

그러나 앞의 8개 특징만으로 모든 마크를 구별하기에는 불충분하다. 실제로 삼각형(▲)과 역삼각형(▼)은 같은 결과 값을 갖기 때문에, 다른 특징을 나타내는 값들이 필요하다. 나머지 9개의 특징은 표 안의 아래 그림과 같이 이미지 상단으로부터 마크까지의 픽셀 수로 정의된다. 따라서 예제에서 제시된 마크의 특징 값은 다음과 같이 정의된다.

$$feature = [b, a + c, 0, 0, d + f, 0, e, g, F_9, F_{10}, F_{11}, \dots, F_{16}, F_{17}]^{(1)}$$

이 특징 값은 0과 1사이로 정규화 되어 신경망에 입력된다.

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

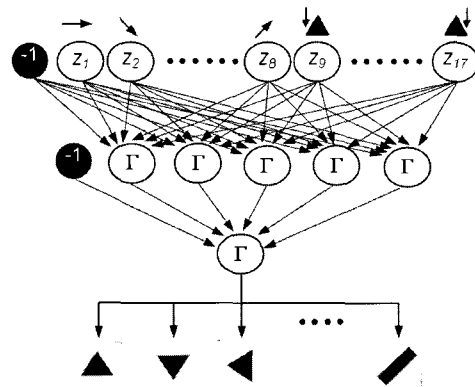


그림 3. 패턴 인식을 위한 신경망 구조.
Fig. 3. Neural networks structure for pattern recognition.

2.2 신경망

패턴인식을 위해 다층퍼셉트론 구조의 역전파(back-propagation) 알고리즘을 적용하였다[5]. 신경망은 그림 4와 같이 18개의 입력층과 6개의 은닉층 1개의 출력층으로 구성한다. Bias는 -1로 설정하고, 활성화 함수는 다음 식을 사용한다.

$$\Gamma(x) = \frac{1}{1 + \exp(x)} \quad (2)$$

IV. 위치 계산을 위한 linear position estimation[3]

Linear position estimation algorithm은 Margrit Betke와 Leonid Gurvits에 의해 이동 로봇 위치 인식(self-localization)을 위해 제안[3]되었고, 기존에 사용한 triangulation 알고리즘이 비선형 방정식들의 최소 제곱해를 이용하기 때문에 연산시간이 오래 걸리는 단점을 극복하기 위하여 제안된 방법이다. 이 방법은 landmark를 복소평면 위의 점으로 간주하여, 해가 p_x, p_y 를 갖는 m 개의 선형 벡터 방정식의 집합으로부터 유도한다.

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

$$z_i^{(r)} = l_i e^{j\tau_i} \text{ for } i = 1, \dots, n \quad (3)$$

와 같이 표현할 수 있다. 여기에서, 길이 l_i 는 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, landmark z_0 와 landmark z_i 사이의 시각(視角, visual angle) φ_i 를 다음과 같이 계산한다.

$$\varphi_i = \tau_i - \tau_0 \text{ for } i = 1, \dots, n \quad (4)$$

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 $v_i^{(r)}$ 그

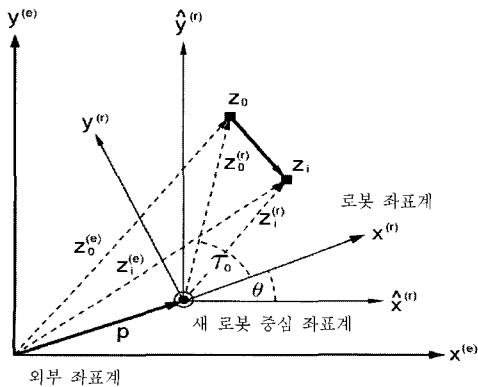


그림 4. Linear estimation algorithm 원리.
Fig. 4. Principle of linear estimation algorithm.

그리고 비 l_i/l_0 로 표현되는 다음과 같은 식을 얻을 수 있다.

$$\frac{1}{z_0^{(r)}} = \frac{l_i}{l_0} \frac{1}{v_i^{(r)}} e^{j\varphi_i} - \frac{1}{v_i^{(r)}} \quad (i = 1, \dots, n) \quad (5)$$

$z_0^{(r)}$ 을 소거하기 위해, (5)의 왼쪽 항을 오른쪽 항의 k 인덱스 식으로 대체하면,

$$\frac{l_k}{l_0} \frac{1}{v_k^{(r)}} e^{j\varphi_k} - \frac{1}{v_k^{(r)}} = \frac{l_i}{l_0} \frac{1}{v_i^{(r)}} e^{j\varphi_i} - \frac{1}{v_i^{(r)}} \quad (i = 1, \dots, n), k \neq i \quad (6)$$

과 같은 식을 얻을 수 있다. (6)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, (6)은 다른 방향을 갖는 로봇중심 좌표계에 대해 다음과 같이 표현될 수 있다.

$$r_k \frac{1}{\widehat{v}_k^{(r)}} e^{j\varphi_k} - \frac{1}{\widehat{v}_k^{(r)}} = r_i \frac{1}{\widehat{v}_i^{(r)}} e^{j\varphi_i} - \frac{1}{\widehat{v}_i^{(r)}} \quad (i = 1, \dots, n), k \neq i \quad (7)$$

여기에서, $r_i = \widehat{z}_i^{(r)} / |\widehat{z}_i^{(r)}|$ 과 $\widehat{v}_i^{(r)} = \widehat{z}_i^{(r)} - \widehat{z}_0^{(r)}$ 만약, 로봇 중심 좌표계를 외부좌표계와 같은 방향으로 설정한다면 로봇 중심 좌표계의 $\widehat{x}^{(r)}$ 와 $\widehat{y}^{(r)}$ 좌표축은 외부 좌표계의 $x^{(e)}$ 와 $y^{(e)}$ 에 각각 평행하게 되며,

$$\widehat{v}_i^{(r)} = v_i^{(e)} = z_i^{(e)} - z_0^{(e)} \quad (8)$$

을 만족하게 된다. 그렇기 때문에, (7)과 (8)로부터 다음 식을 유도할 수 있다.

$$r_k \frac{1}{v_k^{(e)}} e^{j\varphi_k} - \frac{1}{v_k^{(e)}} = r_i \frac{1}{v_i^{(e)}} e^{j\varphi_i} - \frac{1}{v_i^{(e)}} \quad (i = 1, \dots, n), k \neq i \quad (9)$$

(9)에서 알려지지 않은 인수는 오직 비 r_1, \dots, r_n 만 남게 된다. 여기서, $\widehat{x}^{(r)}$ 와 $\widehat{y}^{(r)}$ 축의 로봇 중심좌표계를 이용하는 것이 로봇 자신이 향하고 있는 방향 θ 를 이미 알고 있다는 것을 의미하는 것은 아니다. 하지만, (9)로부터 일단 $\widehat{z}_0^{(r)}$ 을 계산해 내면 다음 식으로부터 로봇이 향하고 있는 방향 θ 를 계산해 낼 수 있다.

$$\theta = \angle(\widehat{z}_0^{(r)}, \widehat{x}^{(r)}) - \tau_0 \quad (10)$$

(9)는 $Ar = c$ 형태의 행렬 형태로 변환될 수 있다. 여기에서 행렬 A 는 복소수 $b_i = (b_{xi}, b_{yi}) = c_i e^{j\varphi_i}$ 로 구성된 $n(n-1) \times n$ 행렬이며, 벡터 $r = (r_1, \dots, r_n)$ 은 벡터 $\widehat{z}_1^{(r)}, \dots, \widehat{z}_n^{(r)}$ 의 아직 알려지지 않은 길이의 비이다. c 는 복

소수 $c_i=(c_x, c_y)=1/\hat{v}_i^{(\theta)}$ 의 차로 구성된 $n(n-1)$ 차의 벡터이다. A 행렬과 c 벡터를 풀어서 정리하면 다음과 같다.

$$A = \begin{bmatrix} b_1 & -b_2 & 0 & 0 & \dots & 0 & 0 \\ b_1 & 0 & -b_3 & 0 & \dots & 0 & 0 \\ b_1 & 0 & 0 & -b_4 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ b_1 & 0 & 0 & 0 & \dots & 0 & -b_n \\ -b_1 & b_2 & 0 & 0 & \dots & 0 & 0 \\ 0 & b_2 & -b_3 & 0 & \dots & 0 & 0 \\ 0 & b_2 & 0 & -b_4 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & b_2 & 0 & 0 & \dots & 0 & -b_n \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -b_1 & 0 & 0 & 0 & \dots & 0 & b_p \\ 0 & -b_2 & 0 & 0 & \dots & 0 & b_n \\ 0 & 0 & -b_3 & 0 & \dots & 0 & b_n \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & -b_{n-1} & b_n \end{bmatrix} \quad (11)$$

$$c = \begin{bmatrix} c_1 - c_2 \\ c_1 - c_3 \\ c_1 - c_4 \\ \vdots \\ c_1 - c_n \\ c_2 - c_1 \\ c_2 - c_3 \\ c_2 - c_4 \\ \vdots \\ c_2 - c_n \\ \vdots \\ c_n - c_1 \\ c_n - c_2 \\ c_n - c_3 \\ \vdots \\ c_n - c_{n-1} \end{bmatrix} \quad (12)$$

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 최소화 하기 위해 최소 제곱법을 이용하여 r 을 구한다.

평균 에러를 다음과 같이 정의한다.

$$E^2 = \|Ar - c\|^2 \quad (13)$$

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.

$$\begin{aligned} \frac{d}{dr} E^2 &= \frac{d}{dr} \|Ar - c\|^2 \\ &= \frac{d}{dr} (r^T A^T A r - r^T A^T c - c^T A r + c^T c) \\ &= 2A^T A r - 2A^T c \end{aligned} \quad (14)$$

즉,

$$0 = 2A^T A r - 2A^T c \quad (15)$$

$$r = (A^T A)^{-1} A^T c \quad (16)$$

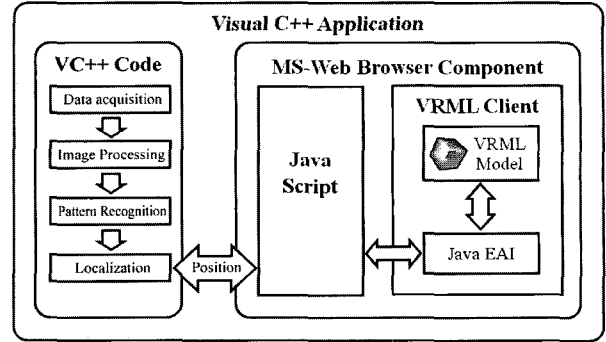


그림 5. Application 구조.
Fig. 5. The structure of the application.

여기서, $A^T A$ 는 nonsingular 행렬이다. (16)로부터 r_i 인자를 갖는 벡터 r 을 계산해 내고, 다음 식을 통해 어떤 임의의 i 에서 $\hat{z}_0^{(i)}$ 를 계산해 낼 수 있다.

$$\frac{1}{\hat{z}_0^{(i)}} = r_i \frac{1}{v_i^{(\theta)}} e^{i\theta_i} - \frac{1}{v_i^{(\theta)}} \quad (17)$$

(17)은 (5)로부터 유도할 수 있으며, 이 식으로부터 일단 $\hat{z}_0^{(i)}$ 이 계산되면 로봇의 위치 p 는

$$p = z_0^{(\theta)} - \hat{z}_0^{(i)} \quad (18)$$

로부터 계산할 수 있다.

V. VRML과의 영상 오버레이

앞 장에서 제시한 Linear Position Estimation 알고리즘에 의해 계산된 로봇의 좌표로봇의 위치(x, y)와 진행방향(θ)를 계산해 낸다. 여기서 계산된 x, y, θ 는 VRML에 적용되어 Wireframe의 VRML 영상을 생성하며, 이 영상은 비전영상에 오버레이 된다.

그림 5와 같이 자바 스크립트(Java Script)와 자바 EAI (External Authoring Interface)를 통해 VRML 모델에 적용된다[6]. EAI는 VRML과 자바애플릿(Java Applet)이 서로 연계될 수 있도록 인터페이스를 제공한 것이며, HTML에 삽입된 자바 애플릿에 의해 VRML의 제어를 용이하게 한다. VC++와 자바 애플릿의 데이터 교환은 자바 스크립트를 통하여 이루어진다.

VI. 시뮬레이션

1. 시뮬레이션 장비

시뮬레이션 실험 장비 사양은 표 2와 같다.

실험을 위해 실제 로봇 작업환경 대신 그림 6과 같이 600×800mm²의 모형을 제작하였다. 모형의 바닥에는 실제 카메라의 위치를 측정할 수 있도록 각도를 측정하기 위한 각도기를 설치하였다. 모형의 왼쪽 하단 모서리를 좌표계의 원점으로 하고, 카메라가 모형을 정면으로 향할 때 방향각을 0°, 카메라가 오른쪽으로 회전하면 (-)각도, 왼쪽으로 향하면 (+)각도로 설정하였다.

표 2. 실험 장비 사양.

Table 2. Specification of the experimental equipment.

장비	특징
카메라 본체	기종 : SDC-411 촬영상자 : 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

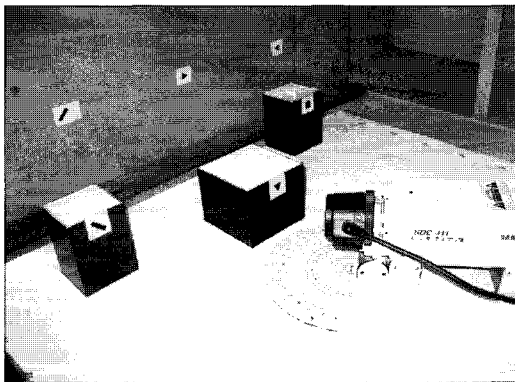
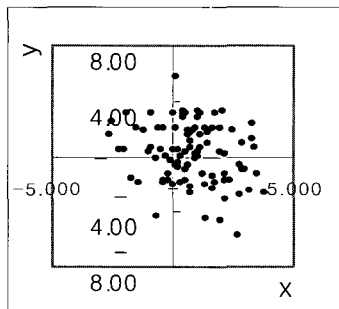
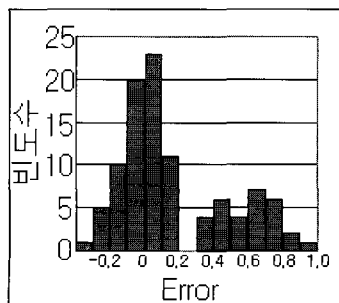


그림 6. Landmark-localization 실험 장치.

Fig. 6. Experimental setup for landmark and localization.



(a) Location and angle



(b) Error

그림 7. 실험 결과.

Fig. 7. Experiment results.

표 3. 위치, 각도 오차 측정 데이터 통계.

Table 3. Statistical result for the location, angle and error.

	x(mm)	y(mm)	$\theta(^{\circ})$
평균	0.758	0.261	0.162
표준편차	1.419	2.995	0.317

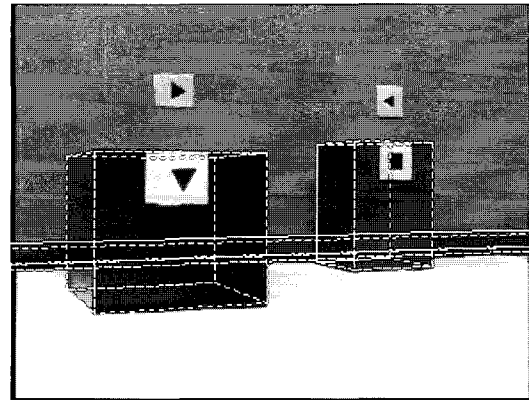


그림 8. 비전영상과 VRML의 오버레이.

Fig. 8. Overlay of the vision image and VRML.

2. Landmark-localization 실험 방법

제작된 로봇 작업환경 모형에 모든 방향에 landmark를 설치해야하나 본 실험에서는 한 쪽 방향에만 landmark를 설치하고 한 방향에 대한 실험만 실시하였다. 카메라를 눈금자 위에 무작위로 위치시키고 landmark를 향한 후 localization을 수행한다. localization 결과와 실제 카메라의 위치를 눈금자로부터 읽어 비교하고 정확도를 계산한다.

실험은 위치와 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.

위치와 방향각 측정 실험은 각각 100회씩 실시하였다.

3. 실험 결과 및 고찰

그림 7은 카메라의 위치와 각도를 측정할 때 발생한 오차분포를 보여준다. 위치 오차는 x,y 좌표 각각 6mm 이내에서 나타났으며, x오차가 2.63mm 이고, y오차가 5.72mm 일때 최대오차가 발생했다. 방향각 에러는 -0.4°에서 1°사이에서 주로 발생했으며 대부분 -0.1°에서 0.1°사이에 분포했다. 실제 로봇의 작업 환경을 6m×8m로 가정했을 때 비례적으로 오차 약 6cm 이내의 오차 범위를 갖는 우수한 localization 성능을 보여 주는 것을 알 수 있다.

표 3에서의 위치 및 각도 오차 측정 데이터 통계에서 방향각 에러가 (+)방향으로 더 많은 빈도수를 갖는 이유는 실험환경 조건 때문으로 판단되며 모델을 설치한 왼쪽의 창문과 조명 때문에 카메라의 자동 조명 조절 장치가 영향을 받아 landmark의 위치 인식에 많은 오차를 포함했기 때문이다.

그림 8은 측정된 위치와 각도 값을 적용하여 구성한 VRML 영상을 실제 비전 영상에 오버레이한 결과이다. 정확한 localization은 그림과 같이 정확한 오버레이로 나타난다.

다. 그러나 두 영상이 불일치는 localization 결과에 애러가 포함되어 있음을 나타내기 때문에 이 경우 오차를 보정하기 위한 프로세스를 수행해야 한다.

VII. 결론 및 향후과제

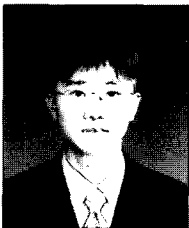
본 논문에서는 landmark를 이용한 위치인식과 위치인식의 오차를 계산된 결과와 VRML 영상과의 오버레이를 구현하는 시스템을 구현하였다.

구현된 시스템은 CCD 카메라로 입력받은 영상으로부터 영상 처리 기법을 통해 landmark들을 추출하고, 이를 신경망 패턴인식 기법을 통해 인식하였다. 인식된 이 landmark의 정보와 landmark 사이 각을 이용해 linear position estimation 알고리즘으로 로봇 위치 인식을 수행하였고, 계산된 로봇의 위치와 방향각을 토대로 3차원 가상 영상을 실현하였다. 이 영상에 구현된 3차원 데이터를 CCD 카메라 영상과 오버레이 하여 그림 8과 같은 영상을 구현하였다. landmark-localization 시스템의 위치 오차는 6mm 이내, 방향 오차는 1° 이내의 결과를 얻었다.

본 연구의 향후과제는 두 영상간의 일치 정도를 각 영상을 구성하는 선이나 교점들의 비교 분석을 통해 매칭율을 개선하는 연구가 필요하다.

참고문헌

- [1] 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.
- [2] 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.
- [3] M. Betke and L. Gurvits, "Mobile robot localization using landmarks," *IEEE Transaction on Robotics and Automation*, vol. 13, no. 2, pp. 251-263 Apr. 1997.
- [4] K. Briechle 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.
- [5] J. M. Zurada, "Introduction to artificial neural system," West Publishing Company.
- [6] M. Rohrmeier "Web based robot simulation using VRML," *Proc. Simulation Conf.*, pp. 1525-1528 vol. 2, Feb. 2000.



권 방 현

2004년 전북대 기계공학과 학사. 2004년~현재 동 대학 석사과정. 관심분야는 로봇 Localization.



손 은 호

2005년 원광대 전기전자공학과 학사. 2005년~현재 전북대학 제어계측공학과 석사과정. 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~2005년 대불대학교 전기전자공학부 전임강사, 조교수. 2005년~

현재 군산대학교 기계공학부 조교수. 관심분야는 메카트로닉스, 재활복지공학, 로봇센싱, Robotics.



정 길 도

1984년 미국 Oregon State University 기계공학학사. 1986년 미국 Georgia Institute of Technology 기계공학석사. 1992년 미국 Texas A&M University 기계공학박사. 1993년~1995년 영남대학교 전임강사. 1995년~현재 전북대학교

전임강사, 부교수, 교수. 관심분야는 Time-Delay, 실시간 멀티미디어 전송, Web 기술, Robotics.