

Compressed Extended Kalman 필터를 이용한 야외 환경에서 주행 로봇의 위치 추정 및 지도 작성

Simultaneous Localization & Map-building of Mobile Robot in the Outdoor Environments by Vision-based Compressed Extended Kalman Filter

윤석준, 최현도, 박성기, 김수현, 곽윤근*
(Sukjune Yoon, Hyun Do Choi, Sung-kee Park, Soo Hyun Kim, and Yoon Keun Kwak)

Abstract : In this paper, we propose a vision-based simultaneous localization and map-building (SLAM) algorithm. SLAM problem asks the location of mobile robot in the unknown environments. Therefore, this problem is one of the most important processes of mobile robots in the outdoor operation. To solve this problem, Extended Kalman filter (EKF) is widely used. However, this filter requires computational power ($\sim O(N)$, N is the dimension of state vector). To reduce the computational complexity, we applied compressed extended Kalman filter (CEKF) to stereo image sequence. Moreover, because the mobile robots operate in the outdoor environments, we should estimate full d.o.f.s of mobile robot. To evaluate proposed SLAM algorithm, we performed the outdoor experiments. The experiment was performed by using new wheeled type mobile robot, Robhaz-6W. The performance results of CEKF SLAM are presented.

Keywords : mobile robot, outdoor exploration, simultaneous localization and map-building, compressed extended kalman filter

1. 서론

21세기 들어 로봇은 인간의 생활 저변에 깊숙하게 관여하게 되었다. 초창기의 로봇은 인간을 대신하여 산업 현장에서 단순 작업을 수행하기 위하여 개발되었다. 그러므로 초기의 로봇은 산업현장의 어느 한 위치에 고정되어서 주어진 기능을 수행하는 존재에 불과하였다. 그러나 점차 산업의 발달과 인간의 생활 수준의 향상으로 인하여 로봇의 기능이 다양해지고, 그 응용분야는 전문화되고 있다.

특히, 로봇의 응용분야는 다양한 환경인식 기술, 제어 기술 및 통신기술의 비약적인 발달로 인해 우주 탐사와 같은 극한 환경의 탐사, 산업의 전분야 및 인간의 생활환경 등에서 매우 전문화되고 다양해지고 있다[1]. 이러한 응용 분야 중 인간이 작업하기 힘든 위험 환경 탐지와 우주 탐사와 같은 극한 환경에서의 작업은 로봇의 중요한 응용 분야이다.

극한 환경에서의 작업을 수행하기 위해서는 주행 로봇을 원하는 곳으로 이동 시킬 수 있는 기술이 요구된다. 원하는 곳으로 주행 로봇을 이동 시키기 위해서는 주행 로봇은 스스로 주변 환경을 인식하고, 주행 로봇이 현재 위치를 알아내며 이 정보를 바탕으로 주행 로봇의 경로를 계획 할 수가 있어야 원하는 곳으로 이동을 할 수가 있다. 그리고 이동 기구부 또한 지형에 적응이 가능하여 험로 주행이 가능해야 된다.

이 중에서 주행 로봇에 있어서 가장 중요한 기술은 로봇 주변의 환경을 인식하여 주행 로봇의 현재 위치를 추정하고

로봇 주변의 지도를 작성하는 것이다[2]. 많은 경우 로봇의 위치를 파악하기 위해서는 로봇 주변의 지도를 이용하였으며, 반대로 로봇의 정확한 위치를 바탕으로 주변 지도를 작성을 하였다. 그러나 미지의 영역에서 작업을 수행하는 로봇은 주변의 지도가 없을 뿐만 아니라 로봇의 위치 또한 알 수가 없다.

그러므로 이러한 환경에서의 응용 로봇은 센서를 통하여 주변 지형을 인식하고 이를 이용하여 주변의 지형 지물을 이용한 지도 작성과 이 지도를 이용한 위치 추정을 동시에 수행되어야 한다. 이러한 작업을 SLAM(Simultaneous Localization And Map-building)[2-8]이라고 하며 1990년대 후반에 들어와 SLAM에 대한 연구가 활성화 되어 최근 많은 발전이 있었다.

$$X = \begin{bmatrix} x \\ y \\ \theta \\ l_1 \\ l_2 \\ \vdots \\ \vdots \\ l_N \end{bmatrix} \quad \Sigma = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{x\theta} & \sigma_{x l_1} & \sigma_{x l_2} & \cdots & \sigma_{x l_N} \\ \sigma_{yx} & \sigma_y^2 & \sigma_{y\theta} & \sigma_{y l_1} & \cdot & \cdots & \sigma_{y l_N} \\ \sigma_{\theta x} & \sigma_{\theta y} & \sigma_{\theta}^2 & \sigma_{\theta l_1} & \cdot & \cdots & \sigma_{\theta l_N} \\ \sigma_{l_1 x} & \sigma_{l_1 y} & \sigma_{l_1 \theta} & \sigma_{l_1}^2 & \sigma_{l_1 l_2} & \cdots & \cdot \\ \sigma_{l_2 x} & \cdot & \cdot & \sigma_{l_2 l_1} & \sigma_{l_2}^2 & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdots & \cdot \\ \sigma_{l_N x} & \cdot & \cdot & \cdot & \cdot & \cdots & \sigma_{l_N}^2 \end{bmatrix} \quad (1)$$

Dissanayake [2]들은 SLAM문제가 EKF (Extended Kalman Filter)를 이용하여 해결이 가능하다는 것을 증명하였다. 식 (1)은 EKF SLAM의 상태 벡터(state vector)와 오차 공분산 행렬 (error covariance matrix) 을 나타낸다. 상태 벡터는 로봇의 현재 위치 및 방향과 사용되는 표식의 위치 값으로 구성이 되며, 오차 공분산 행렬은 상태 벡터의 오차의 분산을 나타낸다. 이들은 주행 로봇의 이동 명령과 센서를 통하여 측정된 값을 이용하여 지속적으로 갱신이 된다. 그러나 EKF

* 책임저자(Corresponding Author)

논문접수 : 2005. 9. 26., 채택확정 : 2006. 2. 27.

윤석준 : 삼성전자(주)(Sukjune.yoon@samsung.com)

최현도 : KAIST 기계공학과 박사과정(chlguseh@kaist.ac.kr)

박성기 : KIST 선임연구원(skee@kist.re.kr)

김수현, 곽윤근 : KAIST 기계공학과 교수

(soohyun@kaist.ac.kr/ykkwak@kaist.ac.kr)

SLAM의 계산량은 상태 벡터의 크기의 제곱에 비례하기 때문에 상태 벡터의 크기가 커지면 실시간 계산이 불가능하다[4].

이러한 EKF SLAM의 문제점을 극복하기 위하여 Guivant 들은 [4] 로봇이 일정 영역에서 반복된 작업을 수행한다고 가정하고, 상태 벡터에 compressed 필터를 적용한 CEKF (Compressed Extended Kalman Filter) SLAM을 제안하였다. 그러나 이 방법은 주행 로봇이 일정한 영역에서 오랜 시간을 작업해야만 적용이 가능하다는 단점이 있다. Montemerlo 들은 [5] 입자 필터(particle filter)와 EKF를 이용하여 상태 벡터의 표식을 분리하여 각 표식에 대하여 독립적으로 EKF를 사용하여 fastSLAM을 제안하였다. 그러므로 fastSLAM의 정확도는 입자 필터의 입자의 개수에 영향을 받는다. 그리고 Thrun 들은[6] 유효한 표식만을 이용한 SEIF(Sparse Information Filter)를 이용하여 실시간 SLAM을 구현하였다. 그러나 이 방법은 정보 행렬(information matrix)을 구하기 위해서는 역행렬을 구하는 과정이 요구되기 때문에, 행렬의 크기가 커지면 계산 시간이 많이 소요되는 단점이 있다.

위에서 언급된 방법들은 모두 레이저 레인지 파인더(laser range finder) 센서를 사용하였다. 그러므로 2차원에 대해서만 적용이 가능하며 SLAM과정에서 가장 어려운 문제 중에 하나인 센서의 데이터 처리(data association)가 복잡하다. 이러한 점을 해결하고자 A. J. Davison 들은 [7,8] 주행 로봇에 액티브 비전(active vision)을 장착하여, 3차원의 표식의 좌표와 로봇의 운동을 이용하여 EKF SLAM을 구현하였다. 그리고 Jeong 들은 [9]은 카메라의 시야를 건물 천장으로 향하게 하여 이를 이용하여 EKF SLAM을 구현하였다. 그러나 앞 선 연구들은 [7-9] 주행 로봇이 평면에서 운동하는 것을 가정 하였기에, 미지의 야외 영역에는 적용이 불가능 하다.

본 논문에서는 위에서 언급된 문제점을 해결하기 위하여 비전 기반 CEKF SLAM을 제안한다. 앞서 언급된 것 같이 비전을 이용하면, 여러 영상에서의 특징점 추출(feature extraction)과 특징점의 추적(feature tracking)이 가능하다. 이를 바탕으로 특징점의 특징값을 이용하여 유효한 특징점과 유효하지 않은 특징점을 나누어 CEKF의 적용이 가능하다. 이러한 방법은 Guivant 들에[4] 의해서 제안된 CEKF SLAM이 가지는 단점(일정 영역에서의 오랜 시간 작업)을 해결 할 수가 있으며, 유효한 특징점을 이용하여 SLAM을 수행하기 때문에, 계산량이 줄어 실시간 처리가 가능하다.

II장에서는 본 논문에 사용된 주행 로봇 Robhaz-6W과 스테레오 비전 시스템에 대하여 기술을 하며, III장은 제안된 비전 기반 CEKF SKAM을 다룬다. 그리고 IV은 제안된 방법을 이용하여 야외 환경에서의 실험 결과를 보여주고 마지막 V 장에서는 결론을 기술한다.

II. 시스템 개요

주행 로봇을 이용하여, 새로운 지형의 탐사를 수행하기 위해서는 험로를 극복할 수 있는 이동 기구부와 주변의 환경을 인식 할 수 있는 시스템이 필요하다. 본 장에서는 주행 로봇으로 사용된 Robhaz-6W(Robot for hazardous application-6 Wheel)과 스테레오 비전 시스템 및 전체적인 시스템에 대하여 기술한다.

1. 주행 로봇 Robhaz-6W

본 연구에서 사용된 주행 로봇은 험로 주행이 가능한 Robhaz-6W이다. Robhaz-6W의 디자인 개념은 “적응성”과 “수동성”이다[10]. 차륜 주행 로봇은 부드러운 평면의 운동에 있어서 좋은 효율을 나타내지만, 험로의 주행 성능은 무한 계도형 등의 주행 로봇에 비해서는 떨어진다. 그러므로 이러한 단점을 보완하고자 Robhaz-6W은 비정형화 되어 있는 지형에 대하여 수동적으로 적응을 하여 주행 성능을 향상 시키는 4 절 링크 구조(4-bar linkage mechanism)를 가진다. 그러므로 험로 주행에 있어서 특별한 제어기법의 도움 없이 Robhaz-6W은 수동적으로 지형에 적응하며 주행할 수가 있다. Robhaz-6W의 기하학적인 구조는 그림 1에 나타나 있으며, 그림 2는 Robhaz-6W이 블록한 지형과 오목한 지형에 대하여 적응하는 모습을 나타낸다.

로봇 몸체에는 싱글 보드 컴퓨터 (SBC: Single Board Computer), 모터 드라이버, 캔 모듈 (CAN: Controller Area Network), 배터리(battery) 및 무선 랜(wireless LAN)등의 로봇 제어부가 장착되어 있다.

2. 스테레오 카메라

미지의 영역을 탐사하기 위해서는 주변의 지형 정보 인식이 필수적이다. 본 연구에서는 이를 위하여 스테레오 비전 카메라를 이용하였다. 야외에서 사용이 되기 때문에 조명 변화에 대하여 둔감한 WDR(Wide Dynamic Range)를 사용하였

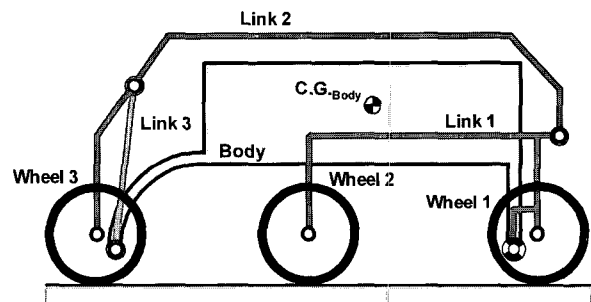


그림 1. Robhaz-6W 구성도.
Fig. 1. The schematic diagram of robhaz-6W.

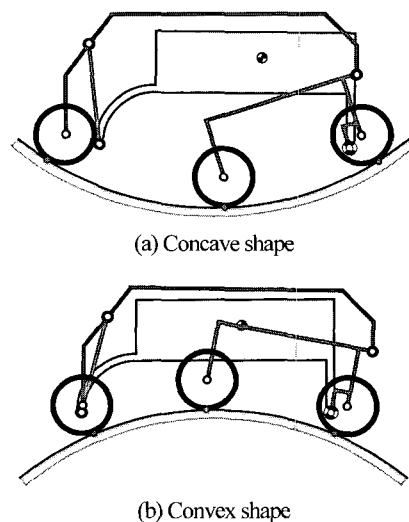
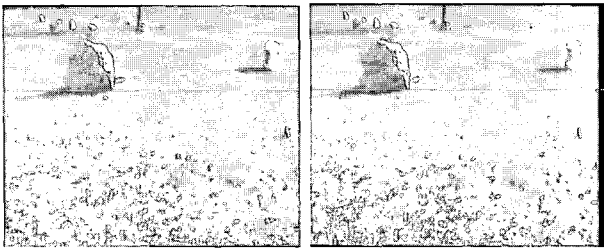


그림 2. Robhaz-6W 지형 적응성.
Fig. 2. The adaptability of robhaz-6W.



(a) Left rectified image (b) Right rectified image

그림 3. 카메라 정화 작업.
Fig. 3. Camera rectification.

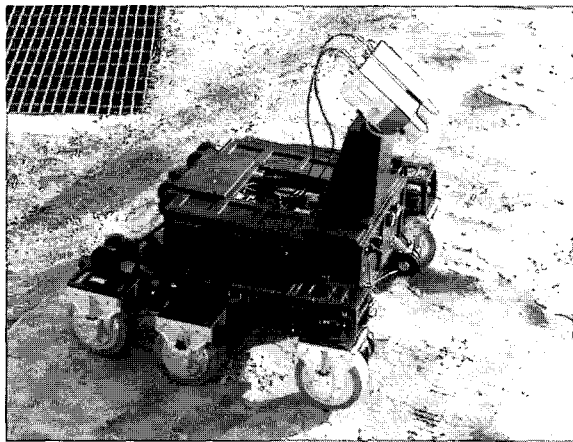


그림 4. Robhaz-6W.
Fig. 4. The prototype of robhaz-6W.

으며, 빛의 양에 따라 조리개를 조절하는 기능이 있는 렌즈(auto iris lens)를 사용하였다.

카메라를 사용하기 위해서는 카메라의 보정(calibration) 및 정규화(rectification) 작업이 필요하다. 본 연구에서는 Zhang, [10]에 의해서 제안된 평면패턴을 이용한 보정 방법을 이용하여 스테레오 카메라의 특성을 추출하였다. 위의 방법을 이용하여 카메라의 특성을 추출한 후 두 대의 카메라의 정규화 작업은 Fusiello 들에 [12] 의해 제안된 방법을 이용하였다.

위의 결과를 이용하여 스테레오 영상의 정규화 작업을 수행하여 아래의 그림 3과 같은 결과를 얻을 수가 있다.

그림 3에 보여지는 것과 같이 두 영상의 동일점(correspondence point)은 같은 y축에 존재하게 되므로, y축을 기준으로 스테레오 매칭을 수행하면 된다.

3. 전체시스템

원격지의 사용자는 Robhaz-6W에게 무선 랜을 이용하여 원하는 목적지의 좌표를 전달해주며, Robhaz-6W은 스테레오 카메라와 자세 센서(3DM sensor)를 이용하여 주변 지형 인식 및 자기 위치를 추정하여 수신된 목적지로 이동을 한다. 경로 계획은 Montano 들에 [13] 의해서 제안된 방법을 사용하였다.

모든 프로그램은 microsoft사의 Visual C++ 6.0에서 작성이 되었으며, 운영체제는 microsoft사의 windows XP이다. 무선 랜을 통하여 전송되는 정보의 손실을 막기 위하여 TCP/IP 통신 프로토콜(protocol)이 사용되었다. Robhaz-6W에 사용된 CPU는

병렬처리를 위하여 SIMD(Single Instruction Multiple Data)가 지원되는 intel사의 Pentium IV 2.2GHz이며, 싱글 보드 컴퓨터는 사용된 CPU를 지원하는 IEI사의 NOVA-8890을 사용하였다. 각 6개의 모터는 CAN 2.0을 통하여 제어가 된다. 그리고 리튬-폴리머(lithium-polymer) 배터리를 사용하였다. 아래의 그림 4는 제작된 Robhaz-6W을 보여준다.

III. 비전 기반 CEKF SLAM

본 장에서는 카메라의 영상을 이용하여, 주행 로봇 Robhaz-6W이 위치하는 곳의 위치 인식에 대하여 기술한다. 로봇의 위치를 알아내기 위해서는 스테레오 입력 영상에서 임의의 특징점(feature)를 추출하여 이 특징점에 CEKF(Compressed Extended Kalman Filter)를 적용하여 특징점의 위치와 함께 추정을 한다. 아래의 그림 5은 비전 기반 CEKF SLAM의 전체적인 흐름을 보여준다.

1. 데이터 처리(data association)

SLAM을 수행하기 앞서 데이터 처리(data association) 작업이 선행 되어야 된다. 영상 기반 CEKF SLAM을 수행하기 위해서는 스테레오 영상으로부터 특징점을 추출하고, 이 점을 유효한 그룹과 유효하지 않은 그룹으로 나누어 주어야 된다. 그리고 이 특징점을 연속되는 영상에서 추적해야 된다.

본 논문에서는 영상에서 특징점을 구해주는 방법으로 Shi-Tomasi 특징점[14]을 이용하였다. Shi-Tomasi 특징점은 계산 시간이 짧으며, 특징점의 고유치(eigen value)의 값이 클수록 좋은 자취 추적 성능을 보인다. 그림 6(a)는 Robhaz-6W의 전방 영상에서 Shi-Tomasi 특징점을 나타낸다. 선택된 특징점의 y축 좌표는 120이하(영상 크기: 320x240)로 정의 하였다. 그리고 그림 6(b)는 선택된 특징점을 Lucas-Kanade[15]에 의해서 제안된 방법으로 좌표 추적한 결과를 보여준다.

Shi-Tomasi 특징점의 고유치의 값이 큰 유효한 특징점은 유효하지 않은 특징점보다 좌표 추적 시 발생하는 오차가 적

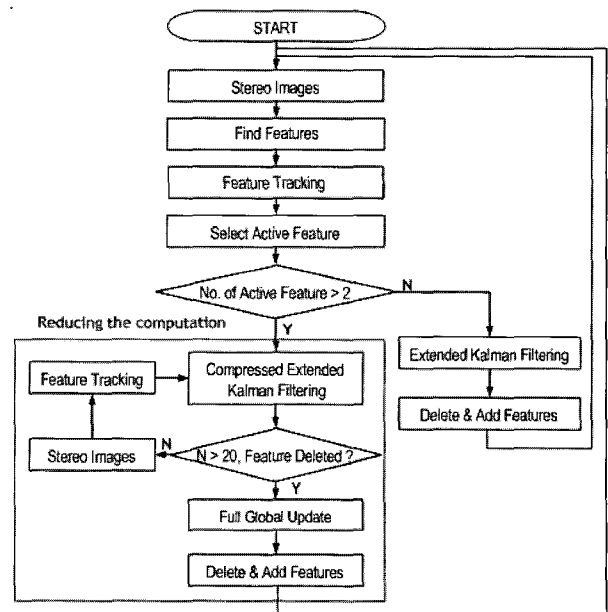


그림 5. 비전 기반 CEKF SLAM 흐름도.
Fig. 5. The flow of vision-based CEKF SLAM.

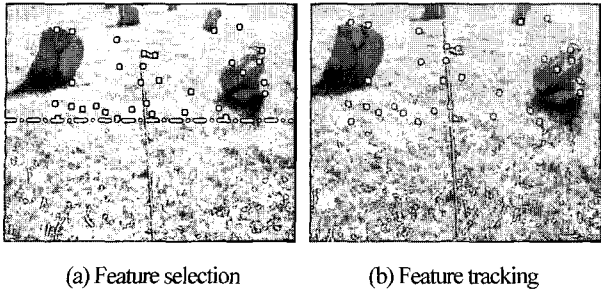


그림 6. 특징점 추출 및 좌표 추적.
Fig. 6. Feature selection & tracking.

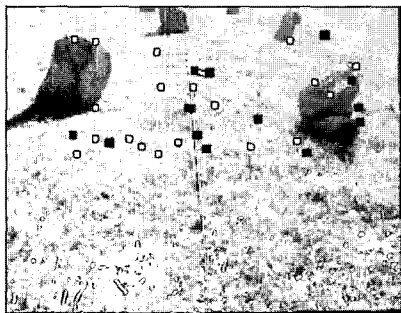


그림 7. 특징점 분류.
Fig. 7. Active & inactive features.

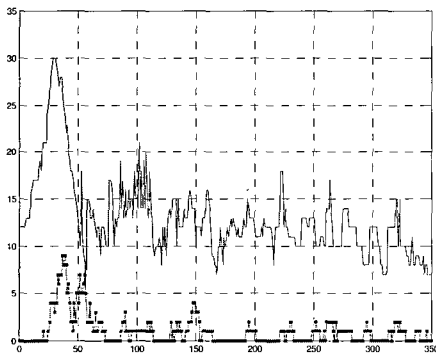


그림 8. 유효한 특징점(붉은색 사각형)과 유효하지 않은 특징점(파란색 점)의 추적 성능.
Fig. 8. The tracking results of active(red square) and inactive(blue point) features.

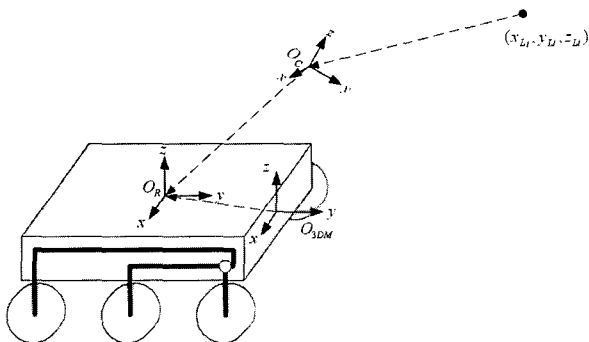


그림 9. Robhaz-6W 좌표계.
Fig. 9. The coordinates of robhaz-6W.

다. 그러므로 고유치가 큰 특징점은 더 유효하게 사용될 수가 있다. 아래의 그림 7의 검은 점은 높은 고유치의 값에 의해 분류된 유효한 특징점이고, 테두리만 있는 흰 점은 유효하지 않은 특징점이다. 그림 8은 이 특징점들의 추적 성능을 나타낸다. 그림 8에 보여지는 것과 같이 고유치에 의해 특징점을 분류할 수가 있다.

2. CEKF SLAM 구현

CEKF는 유효한 값(X_A)에 대해서는 EKF를 이용하여 계속적으로 갱신을 해주고 유효하지 않은 값(X_B)에 대해서는 일정 주기마다 X_A 를 계산할 때 발생하는 부가 행렬(auxiliary matrix)을 이용하여 갱신을 해준다.

(2)는 CEKF SLAM식이고, (3)은 CEKF SLAM에 사용되는 상태 벡터이다. X_A 는 주행 로봇의 위치 및 방향과 유효한 특징점의 좌표이며, X_B 는 유효하지 않은 특징점의 좌표이다. Q_s , Q_r 는 모터와 센서의 오차 공분산 행렬(error covariance matrix), U 행렬은 입력 명령으로 양 바퀴의 속도 V_L , V_R 이다. 그리고 Z 벡터는 센서 출력 값이다.

$$\begin{aligned} \hat{X}_k &= f(X_{k-1}, U_k + Q_r) \\ Z &= h \cdot (X_k) + Q_s \end{aligned} \tag{2}$$

$$X = \begin{matrix} \left. \begin{matrix} x \\ y \\ z \\ \psi \\ \theta \\ \phi \\ x_{A1,1} \\ y_{A1,1} \\ z_{A1,1} \\ \vdots \\ x_{N,1} \\ y_{N,1} \\ z_{N,1} \\ \vdots \end{matrix} \right\} \begin{matrix} \text{Robot Position} \\ \\ \\ \\ \\ \\ \text{Location of Active Features} \end{matrix} \right\} X_A \\ \left. \begin{matrix} x_{B1,1} \\ y_{B1,1} \\ z_{B1,1} \\ \vdots \end{matrix} \right\} \begin{matrix} \text{Location of Inactive Features} \end{matrix} \right\} X_B \end{matrix} \tag{3}$$

CEKF SLAM은 로봇의 운동 명령으로 로봇의 좌표를 예측하는 단계와 센서들의 값을 이용하여 위치를 보정하는 단계로 나누어 진다. 사용된 좌표계는 그림 9에 나타난 것과 같다. 센서로 입력 값은 로봇의 좌표계로 변환하여 사용을 하였다.

1) 운동 예측(motion prediction)

좌우의 속도 명령이 V_L , V_R 이고 운동 명령의 시간차이가 dt 일 경우, Robhaz-6W의 위치는 아래의 (4)과 같다. (4)에서 ψ , θ , ϕ 는 roll, pitch, yaw 그리고 L 은 Robhaz-6W의 두 바퀴 사이의 길이이다.

$$\begin{aligned} \hat{x}_k &= x_{k-1} + dt \cdot \left\{ \frac{V_L + V_R}{2} (-\sin \phi \cos \psi + \cos \phi \sin \theta \sin \psi) \right\} \\ \hat{y}_k &= y_{k-1} + dt \cdot \left\{ \frac{V_L + V_R}{2} (\cos \phi \cos \psi + \sin \phi \sin \theta \sin \psi) \right\} \\ \hat{z}_k &= z_{k-1} + dt \cdot \left\{ \frac{V_L + V_R}{2} \cos \theta \sin \psi \right\} \end{aligned} \tag{4}$$

$$\begin{aligned} \hat{\psi}_k &= \psi_{k-1} + dt \cdot \left\{ \frac{V_R - V_L}{L} (\sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi) \right\} \\ \hat{\theta}_k &= \theta_{k-1} + dt \cdot \left\{ \frac{V_R - V_L}{L} (-\cos \phi \sin \psi + \sin \phi \sin \theta \cos \psi) \right\} \\ \hat{\phi}_k &= \phi_{k-1} + dt \cdot \left\{ \frac{V_R - V_L}{L} \cos \theta \cos \psi \right\} \end{aligned}$$

그리고 EKF의 상태 벡터의 오차 공분산 행렬을 갱신하는데 사용되는 Robhaz-6W의 자코비안(Jacobian)은 (5), (6)과 같다.

$$\frac{\partial F}{\partial X} = \begin{bmatrix} 1 & 0 & 0 & dt \cdot \frac{V_R + V_L}{2} \cdot (\sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi) \\ 0 & 1 & 0 & dt \cdot \frac{V_R + V_L}{2} \cdot (-\cos \phi \sin \psi + \sin \phi \sin \theta \cos \psi) \\ 0 & 0 & 1 & dt \cdot \frac{V_R + V_L}{2} \cdot (\cos \theta \cos \psi) \\ 0 & 0 & 0 & 1 + dt \cdot \frac{V_R - V_L}{L} \cdot (\sin \phi \cos \psi - \cos \phi \sin \theta \sin \psi) \\ 0 & 0 & 0 & dt \cdot \frac{V_R - V_L}{L} \cdot (-\cos \phi \cos \psi - \sin \phi \sin \theta \sin \psi) \\ 0 & 0 & 0 & -dt \cdot \frac{V_R - V_L}{L} \cdot (\cos \theta \sin \psi) \end{bmatrix} \quad (5)$$

$$\begin{aligned} & \left. \begin{aligned} & dt \cdot \frac{V_R + V_L}{2} \cdot (\cos \phi \cos \theta \sin \psi) & dt \cdot \frac{V_R + V_L}{2} \cdot (-\cos \phi \cos \psi - \sin \phi \sin \theta \sin \psi) \\ & dt \cdot \frac{V_R + V_L}{2} \cdot (\sin \phi \cos \theta \sin \psi) & dt \cdot \frac{V_R + V_L}{2} \cdot (-\sin \phi \cos \psi + \cos \phi \sin \theta \sin \psi) \\ & dt \cdot \frac{V_R + V_L}{2} \cdot (-\sin \phi \sin \psi) & 0 \\ & dt \cdot \frac{V_R - V_L}{L} \cdot (\cos \phi \cos \theta \cos \psi) & dt \cdot \frac{V_R - V_L}{L} \cdot (\cos \phi \sin \psi - \sin \phi \sin \theta \cos \psi) \\ & 1 + dt \cdot \frac{V_R - V_L}{L} \cdot (\sin \phi \cos \theta \cos \psi) & dt \cdot \frac{V_R - V_L}{L} \cdot (\sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi) \\ & dt \cdot \frac{V_R - V_L}{L} \cdot (-\sin \theta \cos \psi) & 0 \end{aligned} \right\} \\ & \frac{\partial F}{\partial U} = \begin{bmatrix} 0.5 \cdot dt \cdot (-\sin \phi \cos \psi + \cos \phi \sin \theta \sin \psi) \\ 0.5 \cdot dt \cdot (\cos \phi \cos \psi + \sin \phi \sin \theta \sin \psi) \\ 0.5 \cdot dt \cdot (\cos \theta \sin \psi) \\ -\frac{dt}{L} \cdot (\sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi) \\ -\frac{dt}{L} \cdot (-\cos \phi \sin \psi + \sin \phi \sin \theta \cos \psi) \\ -\frac{dt}{L} \cdot (\cos \theta \cos \psi) \\ 0.5 \cdot dt \cdot (-\sin \phi \cos \psi + \cos \phi \sin \theta \sin \psi) \\ 0.5 \cdot dt \cdot (\cos \phi \cos \psi + \sin \phi \sin \theta \sin \psi) \\ 0.5 \cdot dt \cdot (\cos \theta \sin \psi) \\ \frac{dt}{L} \cdot (\sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi) \\ \frac{dt}{L} \cdot (-\cos \phi \sin \psi + \sin \phi \sin \theta \cos \psi) \\ \frac{dt}{L} \cdot (\cos \theta \cos \psi) \end{bmatrix} \quad (6) \end{aligned}$$

2) 측정값을 이용한 위치 보정(measurement update)
Robhaz-6W의 이동 명령 값을 이용하여 위치를 추정한 후,

스테레오 카메라와 자세 센서(3DM)의 측정값을 이용하여 위치를 보정한다. 스테레오 알고리즘은 Yoon들에 [16]의해서 제안된 실시간 알고리즘을 사용하였고, 스테레오 카메라의 센서 모델은 Matthise [17]에 의해서 제안된 방법을 사용하였다.

(2)의 Z은 아래의 (7)과 같다. ψ, θ, ϕ 는 roll, pitch, yaw이며, ${}^R x_{Li}, {}^R y_{Li}, {}^R z_{Li}$ 는 로봇의 좌표 기준 측정된 특징점의 3차원 좌표값이다.

$$Z = \begin{bmatrix} \psi \\ \theta \\ \phi \\ {}^R x_{Li} \\ {}^R y_{Li} \\ {}^R z_{Li} \\ \vdots \\ \vdots \end{bmatrix} \quad (7)$$

스테레오 카메라에 의해서 얻어지는 각 특징점의 좌표는 아래의 (8)에 의해서 로봇 좌표계로 변환이 된다.

$${}^C_R T = \begin{bmatrix} 0.9999 & 0.0121 & -0.0015 & 56.35 \\ 0.0062 & -0.4005 & 0.9163 & 202.61 \\ 0.0105 & -0.9162 & -0.4005 & 251.22 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

그림 6에서 3DM센서와 스테레오 카메라에 의해 측정되는 $\psi, \theta, \phi, {}^R x_{Li}, {}^R y_{Li}, {}^R z_{Li}$ 는 아래 (9)와 같다,

$$\begin{aligned} roll &= \psi \\ pitch &= \theta \\ yaw &= \phi \end{aligned} \quad (9)$$

$$\begin{aligned} {}^R x_i &= -(z_{Li} - z) \cdot \sin \theta + (x_{Li} - x) \cdot \cos \theta \cos \phi + (y_{Li} - y) \cdot \cos \theta \sin \phi \\ {}^R y_i &= (z_{Li} - z) \cdot \sin \psi \cos \theta + (x_{Li} - x) \cdot [\sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi] \\ & \quad + (y_{Li} - y) [\cos \psi \cos \phi + \sin \psi \sin \theta \sin \phi] \\ {}^R z_i &= (z_{Li} - z) \cdot \cos \psi \cos \theta + (x_{Li} - x) \cdot [\cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi] \\ & \quad + (y_{Li} - y) [-\sin \psi \cos \phi + \cos \psi \sin \theta \sin \phi] \end{aligned}$$

그림 7에 나타난 것과 같이, 유효한 특징점에 대하여 20회 동안 EKF를 적용하여 SLAM을 수행하고, 매 20회 마다 한 번씩 모든 특징점에 대하여 SLAM을 수행한다. 이렇게 CEKF를 적용을 하면, 계산 시간의 단축 효과를 얻을 수 있다.

CEKF는 유효한 특징점에 대한 값만을 이용하기 때문에, 출력 함수와 상태 벡터의 공분산 행렬은 다음의 (10)와 같다. 출력 함수는 (9)에 의해서 얻어진다.

$$\begin{aligned} H &= \frac{\partial h}{\partial X} \Big|_{X=X(t)} = \begin{bmatrix} \frac{\partial h}{\partial X_A} & \frac{\partial h}{\partial X_B} \end{bmatrix} = [H_a \quad 0] \\ P &= \begin{bmatrix} P_{aa} & P_{ab} \\ P_{ba} & P_{bb} \end{bmatrix} \end{aligned} \quad (10)$$

k_1 에서 k_2 시간 동안($k_1 < t < k_2$)를 수행하면, 유효한 특징점에 대한 값인 X_a, P_{aa} 는 지속적으로 EKF에 의해서 갱신이

되고, 그 외의 $X_b, P_{ab}, P_{ba}, P_{bb}$ 는 CEKF계산이 끝날 때(20회 계산 후, 6-8초) 갱신이 된다. 최종적인 P_{ab}, P_{ba}, P_{bb} 는 (13)과 같으며, 이 값들을 구하기 위해서 계산되는 필요한 부가 행렬은 다음의 (11), (12)와 같다. 부가 행렬의 초기값은 (10)에 나타나있다.

Initial Condition

$$\begin{aligned} \phi(k_1) &= I \\ \psi(k_1) &= 0 \\ \theta(k_1) &= 0 \end{aligned} \tag{10}$$

Prediction Step

$$\begin{aligned} \phi(k) &= J_{aa} \cdot \phi(k-1) \\ \psi(k) &= \psi(k-1) \\ \theta(k) &= \theta(k-1) \end{aligned} \tag{11}$$

Update Step

$$\begin{aligned} \phi(k) &= (I - \mu(k)) \cdot \phi(k-1) \\ \psi(k) &= \psi(k-1) + \phi^T(k-1) \cdot \beta(k) \cdot \phi(k-1) \\ \theta(k) &= \theta(k-1) + \phi^T(k-1) \cdot H_a^T(k-1) \cdot S^{-1}(k-1) \cdot z(k-1) \\ \beta(k) &= H_a^T(k) \cdot S^{-1}(k) \cdot H_a(k) \\ \mu(k) &= P_{aa}(k) \cdot \beta(k) \\ S(k) &= H_a^T(k) \cdot P_{aa}(k) \cdot H_a(k) + R \\ z(k) &= y(k) - h(X_a(k), k) \\ P_{ab}(k_2) &= \phi(k_2) \cdot P_{ab}(k_1) \\ P_{bb}(k_2) &= P_{bb}(k_1) - P_{ba}(k_1) \cdot \psi(k_2) \cdot P_{ab}(k_1) \\ X_b(k_2) &= X_b(k_1) - P_{ba}(k_1) \cdot \theta(k_2) \end{aligned} \tag{12}$$

IV. 비전 기반 CEKFSLAM 실험 결과

CEKF SLAM을 Robhaz-6W에 적용하여 성능을 평가하는 실험을 하였으며, 이를 위한 실제적인 야외 실험 환경의 모습은 그림 10과 같다. 이 실험에서 출발점의 좌표는 (0,0,0)이

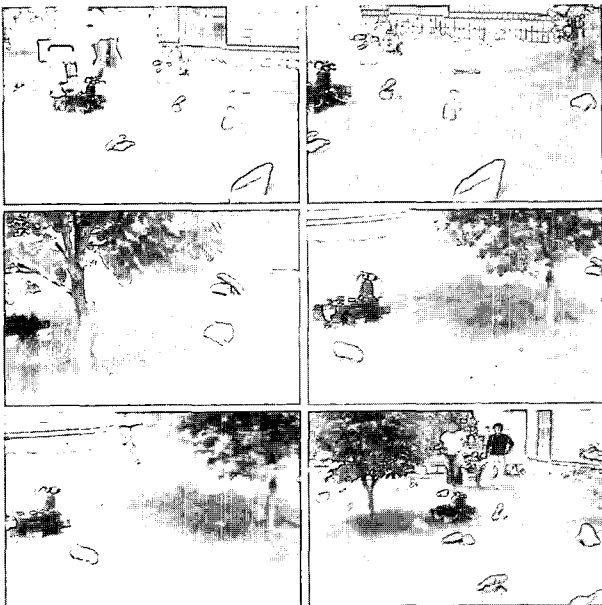
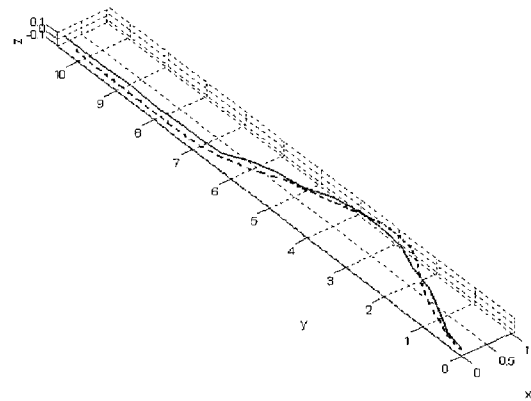
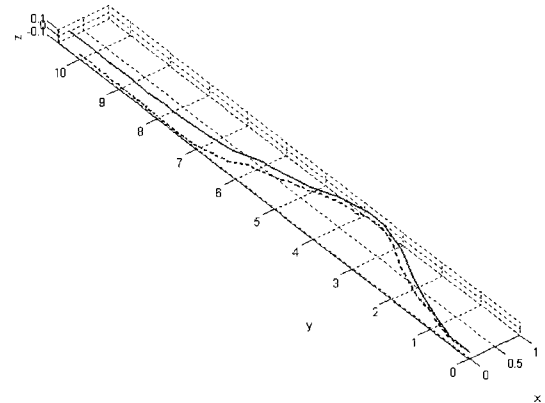


그림 10. 야외 실험 환경.
Fig. 10. The environment of outdoor experiments.



(a) EKF SLAM result



(b) CEKF SLAM result

그림 11. 야외 환경에서의 SLAM 결과.
Fig. 11. SLAM result in the outdoor environment.

표 1. 야외 환경에서의 SLAM 결과.

Table 1. SLAM result in the outdoor environment.

No	EKF SLAM	CEKF SLAM
1	(0.15, 10.50, 0)	(0.21, 10.51, 0)
2	(0.18, 10.59, 0)	(0.19, 10.55, 0)
3	(0.14, 10.54, 0)	(0.23, 10.58, 0)
4	(0.20, 10.58, 0)	(0.20, 10.55, 0)
5	(0.10, 10.48, 0)	(0.15, 10.53, 0)

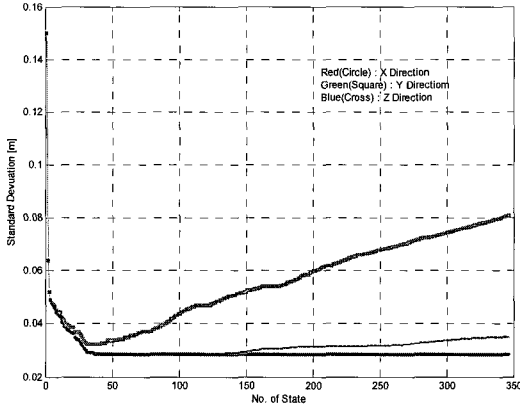
고, 최종 목적지의 좌표는 (0,10,0)이다. 본 실험에서는 실시간 센서입력을 이용하여 경로 계획이 가능한 Montano 들에 [13] 의해서 제안된 방법을 사용하였다.

그림 11(a), (b)는 CEKF 와 EKF를 이용한 SLAM의 결과를 각각 나타낸다. 실선은 Robhaz-6W의 실제 이동 경로이고, 점선은 각 방법에 의해서 추정된 경로이다. 표 1은 y축의 방향으로 10m 주행 실험한 결과를 보여준다. Robhaz-6W의 실제 이동 경로는 Robhaz-6W의 경로 표시 장치에 의해서 지표면에 표시되는 자취를 측정하여 얻었다.

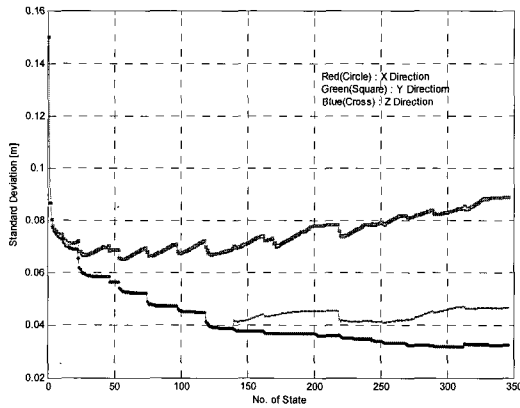
위의 결과를 분석해 보면, EKF SLAM의 경우 10m이 주행 시 약 0.4m의 위치 오차를 가지며, CEKF SLAM의 경우는 약 0.5m의 위치 오차를 가진다. EKF SLAM과 CEKF SLAM의

Robhaz-6W의 (x,y,z) 방향으로의 공분산 오차는 그림 13과 같다. 그림 13에 보여지는 것과 같이 CEKF SLAM의 경우 오차가 갑자기 줄어드는 곳은 전체적으로 갱신이 일어난 경우이다. 그리고 이 경우 특징점의 오차는 그림 14에 보여지는 것과 같이 줄어 들게 된다.

그리고 각 1회에 대한 EKF SLAM과 CEKF SLAM의 계산 시간은 아래의 표 2와 같다.



(a) EKF SLAM result



(b) CEKF SLAM result

그림 12. Robhaz-6W 자세 오차.

Fig. 12. The standard deviation of robhaz-6W in SLAM process.

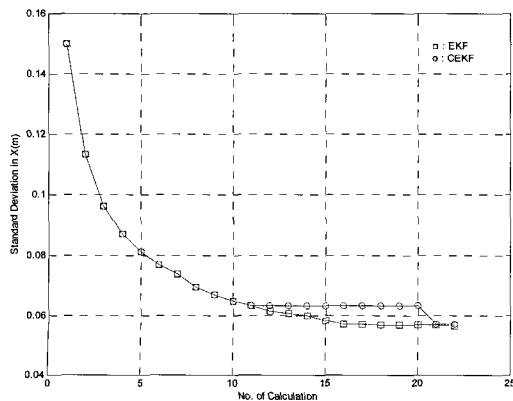


그림 13. CEKF SLAM의 특징점 오차.

Fig. 13. The position error of feature in CEKF SLAM.

표 2. SLAM 계산 시간.

Table 2. Calculating time in SLAM process(msec).

	Calculation Time(msec)
EKF SLAM	92
CEKF SLAM	31

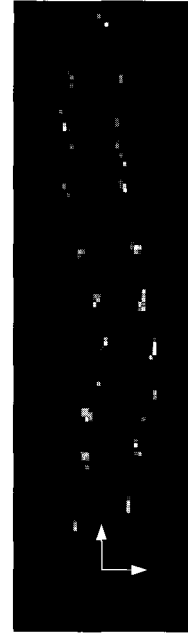


그림 14. 장애물 지도.

Fig. 14. The generated obstacle map.

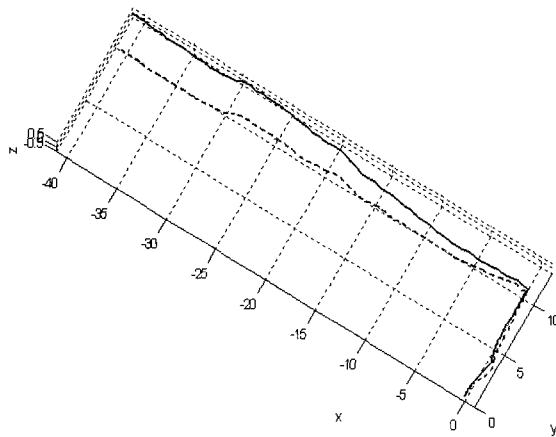
표 1의 결과로부터, EKF SLAM과 CEKF SLAM은 비슷한 위치 추정 정확도를 가지는 것을 알 수가 있다. 그러나 계산식의 단순화로 인해 계산 시 소요되는 시간은 CEKF SLAM이 EKF SLAM의 1/3정도인 것을 알 수가 있다.

아래의 그림 14는 CEKF SLAM의 결과와 스테레오 비전의 3차원 지형 복원을 이용하여 작성된 장애물 지도이다. 흰 부분이 Robhaz-6W이 주행할 수 없는 장애물을 나타낸다.

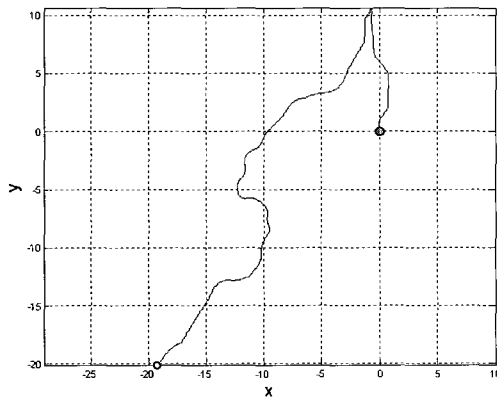
위의 CEKF SLAM을 이용하여 긴 거리의 주행을 수행하였다. 시작점은 (0,0,0), 중간 경유점은 (0,10,0), 최종 목적지의 값은(-40,10,0)이다. 총 주행 거리는 50m이다.

CEKF SLAM을 이용한 위치 추정 결과는 아래의 그림 15(a)에 나타나있다. 그림 15(a)에 점선은 CEKF SLAM에 의해서 추정된 경로이고, 실선은 실제의 이동 경로를 나타낸다. 그리고 그림 15(b)는 Robhaz-6W 모터의 엔코더 값을 이용하여 추정한 경로이다. CEKF SLAM을 이용하여 최종 목적지로 이동 명령을 주었을 경우, Robhaz-6W의 최종 목적지의 좌표는 (-41.8, 12.9, 0.0)으로 약 3.57m의 위치 추정 오차가 발생된다. 그리고 이 경우 모터의 엔코더 값을 이용하여 Robhaz-6W의 위치를 추정하면, (-15.58, -23.41, 0)이 된다. 그림 15(c)는 작성된 장애물 지도이다.

위의 결과로부터 본 논문에서 제안된 비전 기반 CEKF SLAM은 빠른 계산이 가능하며, EKF SLAM과 비슷한 정확도를 가진다. 위치 추정 결과는 10m 주행 시 약 0.5m의 오차를 가지며, 50m 주행 시 약 3.5m의 위치 오차를 가진다.



(a) Result of CEK SLAM



(b) Results of only odometry



(c) The generated obstacle map

그림 15. 긴 거리 주행 시 robhaz-6W의 위치 추정 결과.
Fig. 15. The position estimation result of long range travel.

V. 단위

본 연구에서는 험로 주행이 가능한 주행 로봇에 스테레오 비전 시스템을 장착하여, 미지의 탐사를 수행하기 위해 필수적으로 요구되는 주행 로봇의 위치 추정 기술들에 대하여 기술하였다.

본 연구에 제작된 주행 로봇 'Robhaz-6W'은 4절 링크로 이루어져, 4절 링크가 가지는 1자유도로 지형에 적응을 하여 좋은 주행성능을 나타낸다. 그리고 주행 로봇 주변의 지형 정보의 인식을 위하여 스테레오 비전 시스템을 사용하였다.

그리고 로봇의 위치 및 방향을 파악하기 위해 3차원 지형 정보와 카메라로부터 입력되는 영상에 CEKF(Compressed Extended Kalman Filter)를 적용하여 SLAM(Simultaneous Localization And Map-building)을 구현하였다. 카메라의 입력 영상에서 각 특징점의 중요도를 산출하여, 이 특징점에 CEKF를 적용함으로

위치 산출 시 요구되는 계산량을 줄여주었다. EKF SLAM과 비교하여 계산량이 35% 이하로 줄어 드는 것을 알 수가 있다. 위의 3차원 지형 정보 인식과 주행 로봇의 위치 산출 결과를 이용하여 주행 로봇이 지나간 곳에 대한 장애물 지도를 작성하였다. 제안된 CEKF SLAM의 위치 오차는 10m 주행 시 약 0.5m, 50m 주행시 약 3.5m이다.

이러한 오차는 거리 센서(range sensor)만을 이용할 경우 불가피하게 발생이 된다. 이러한 오차를 줄이기 위해서는 GPS(Global Positioning System) 또는 DGPS(Differential Global Positioning System)과 같은 외부의 센서를 사용해야만 한다. 현재 사용되는 DGPS의 위치 정확도가 약 10m 인 것을 고려하면, 본 논문에서 제안된 CEKF SLAM은 DGPS와 함께 효율적으로 사용이 될 수 있다.

참고문헌

- [1] T. B. Sheridan, *Telerobotics, Automation, and Human Supervisory Control*, MIT Press, Cambridge, 1992.
- [2] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Trans. on Robotics and Automation*, vol. 17, no. 3, pp. 229-241, 2001.
- [3] P. Newman, J. Leonard, J. D. Tardos, J. Neira, "Explore and return: experimental validation of real-time concurrent mapping and localization," in *Proc., IEEE Int. Conf. on Robotics and Automation*, vol. 2, pp. 1802-1809, 2002.
- [4] J. E. Guivant and E. M. Nebot, "Optimization of the simultaneous localization and map-building algorithm for real-time implementation," *IEEE Trans. on Robotics and Automation*, vol. 17, no. 3, pp. 242-257, 2001.
- [5] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *Proc., AAAI Nat. Conf. on Artificial Intelligence*, pp. 593-598, 2002.
- [6] S. Thrun, Y. Liu, D. Koller, A. Y. Ng, Z. Ghahramani, and H. Urrant-Whyte, "Simultaneous localization and mapping with sparse extended information filters," *Int. J. of Robotics Research*, pp. 693-717, 2004.
- [7] A. J. Davison and D. W. Murray, "Simultaneous localization and map-building using active vision," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 865-880, 2002.
- [8] A. J. Davison and N. Kita, "3D simultaneous localisation and map-building using active vision for a robot moving on undulating terrain," in *Proc., IEEE Int. Conf. on Computer Vision and Pattern Recognition*, vol. 1, pp. 384-391, 2001.
- [9] W. Jeong and K. M. Lee, "CV-SLAM: A new ceiling vision-based SLAM technique," in *Proc., IEEE Int. Conf. on Intelligent Robots and Systems*, pp. 3070-3075, 2005.
- [10] C.-H. Woo, C.-H. Cho, S.-C. Kang, K.-C. Shin, M.-S. Kim, S. H. Kim, and Y. K. Kwak, "A new wheeled mobile robot capable of climbing the stairs using a passive linkage mechanism," in *Proc. ASME Int. Conf. on Engineering Systems Design and Analysis, ESDA2002/APM075*, 2002.
- [11] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330-1334, 2000.

[12] A. Fusiello, E. Trucco, and A. Verri, "A compact algorithm for rectification of stereo pairs," *Machine Vision and Applications*, vol. 12, pp. 16-22, 2000.

[13] L. Montano and J. R. Asensio, "Real-time robot navigation in unstructured environments using a 3D laser rangefinder," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 2, pp. 526-532, 1997.

[14] S. Jianbo and C. Tomasi, "Good features to track," in *Proc.*

IEEE Int. Conf. on Computer Vision and Pattern Recognition, pp. 593-600, 1994.

[15] J.-Y. Bouguet, "Pyramidal implementation of the Lucas Kanade feature tracker description of the algorithm," *Tech. Rep., Microprocessor Research Labs*, Intel Corporation, 1999.

[16] S. Yoon, S.-K. Park, S. Kang, and Y. K. Kwak, "Fast correlation-based stereo matching with the reduction of systematic errors," *Pattern Recognition Letter*, vol. 26, pp. 2221-2231, 2005.



윤석준

1999년 KAIST 기계공학부 졸업. 2001년 동 대학원 기계공학과 졸업(공학석사). 2005년 동 대학원 기계공학과 졸업(공학박사). 관심분야는 주행로봇, SLAM, stereo vision.



최현도

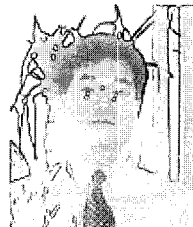
2002년 부산대학교 기계공학부 졸업. 2004년 KAIST 기계공학과 대학원 졸업(공학석사). 2004년~현재 동 대학원 기계공학과 박사과정 재학중. 관심분야는 기구설계, 주행로봇, traction 제어.



박성기

1987년 서울대학교 기계설계학과 졸업. 1989년 동 대학원 기계설계학과 졸업(공학석사). 2000년 KAIST 자동화 및 설계공학과 졸업(공학박사). 2000년~현재 한국과학기술연구원 선임연구원 재직중. 관심분야는 컴퓨터 비전, 로봇 비전,

지능로봇.



김수현

1978년 서울대학교 기계공학과 졸업. 1980년 KAIST 기계공학과 졸업(공학석사). 1991년 Imperial College of Science and Technology, University of London 졸업(공학박사). 1991년~현재 KAIST 기계공학과 교수 재직중. 관심분야는 micro-

actuation, nano 운동 측정과 제어를 위한 광학장치.



곽윤근

1971년 서울대학교 기계공학과 졸업. 1974년 University of Colorado, Boulder, USA 공학설계 및 경제평가학과 졸업(공학석사). 1978년 University of Texas, Austin, USA 기계공학과 졸업(공학박사). 1978년~현재 KAIST 기계공학과 교수

재직중. 관심분야는 지능 로봇, 감성로봇, 경찰로봇.