

Structured Light 기법을 이용한 이동 로봇의 상대 위치 추정 알고리즘 연구

A Study on the Relative Localization Algorithm for Mobile Robots using a Structured Light Technique

노 동 기*, 김 곤 우, 이 범 희
(Dong-Ki Noh, Gon-Woo Kim, and Beom-Hee Lee)

Abstract : This paper describes a relative localization algorithm using odometry data and consecutive local maps. The purpose of this paper is the odometry error correction using the area matching of two consecutive local maps. The local map is built up using a sensor module with dual laser beams and USB camera. The range data from the sensor module is measured using the structured lighting technique (active stereo method). The advantage in using the sensor module is to be able to get a local map at once within the camera view angle. With this advantage, we propose the AVS (Aligned View Sector) matching algorithm for correction of the pose error (translational and rotational errors). In order to evaluate the proposed algorithm, experiments are performed in real environment.

Keywords : structured light, relative localization, laser, mobile robot

I. 서론

Map building은 여러 가지 sensor정보를 가공하는 동적인 과정을 통하여 수행된다. 이때 각각의 sensor는 외부 환경의 물리적인 특징을 검출해내게 된다. 일반적으로 이러한 정보들은 이동로봇의 경로를 따라 얻게 되는데 이때 로봇의 기구학적 특징과 센서 특징에 대한 연구가 필요하게 된다. 또한 로봇의 경로에 따라 sensor정보를 얻게 되므로 로봇의 경로 변화에 대한 정확한 정보가 map building과정 시 필요하게 된다. 각각의 서로 다른 위치에서 sensor정보를 얻기 때문에 서로 다른 위치에서 얻은 sensor정보를 연결하는 것이 중요하다[1]. 단순히 각각의 위치에서 얻은 정보를 dead-reckoning 기법에 의해 연결할 수도 있다. 하지만 encoder정보를 이용한 이동 로봇의 위치와 실제 위치 사이에 항상 오차가 존재하기 때문에 두 개의 연속되는 sensor 정보간의 관계를 알기 위해서는 로봇의 정확한 위치정보를 추정할 필요가 있다. 따라서 sensor 정보를 이용하여 로봇의 위치정보를 추정하기 위한 연구가 많이 이루어 지고 있다.

그 중 하나는 초음파 센서의 검출 영역과 로봇의 이동 정보를 이용하여 로봇 이동 정보를 추정하는 방법이다[2]. 이 연구는 최근에 얻어진 4개의 frame에서의 sensor정보를 융합하여 현재 로봇의 위치를 추정하는 것이다. 하지만 이 방식은 초음파 센서의 검출 영역 내에서도 sensor 정보의 신뢰도가 각각 다르기 때문에 각각의 센서 정보를 융합하는 것이 어렵다는 단점이 있다. 또 하나는 landmark를 이용하여 로봇의 좌표를 추정하는 방법이다[3]. 이것은 절대적 위치 추정 기법(absolute localization)으로 로봇 좌표의 오차가 항상 일정

범위 안에 있다는 장점이 있지만 동적인 환경에서 사용할 수 없다는 단점이 있다. 본 논문에서는 두 개의 laser beam과 vision sensor를 이용하여 얻어지는 2차원적인 정보를 이용한다. 그리고 각각의 위치에서 얻어진 연속된 두 개의 local map을 이용하여 상대적인 위치 추정(relative localization) 기법을 제안한다. 연속된 두 개의 local map과 odometry 정보를 이용하여 로봇의 odometry 오차를 보정하여 상대적인 위치를 추정하는 방법을 제안한다. Map 정보를 이용한 로봇의 좌표 추정에 관한 연구는 Gonzalez 에 의해 제안된바 있다[4]. 또한 많은 연구에서 초음파 sensor의 error특징을 보완하는 알고리즘 연구를 수행해 왔다[5].

본 논문에서는 laser beam과 vision sensor를 이용한 센서 모듈을 사용하여 sensor자체가 가지고 있는 오차를 최소화하였다. Laser beam과 vision sensor를 사용하기 때문에 vision sensor의 특징인 시야 각(view angle)에 의해 vision sensor가 검출할 수 있는 영역(view sector)을 정의할 수 있다[6]. 이때 두 개의 연속된 local map에서 동일한 검출 영역(view sector)을 정의할 수 있는데 이것을 AVS (Aligned View Sector)라고 정의한다.

본 논문에서는 연속되는 두 개의 local map에 대한 AVS 영역의 matching을 이용하여 odometry 오차를 보정하여 상대적인 위치 추정(relative localization)을 하고 이를 이용하여 보다 정확한 global map을 작성하는 방법을 제안한다. 또한 두 개의 laser beam에 의해 생성된 각각의 local map을 이용하여 위치 추정의 신뢰성을 향상시키는 방법을 제안한다.

II. 센서 모듈

센서 모듈을 탑재한 mobile robot시스템 구성은 그림 1(a)와 같다. 장애물과의 거리 측정은 그림 1(b)에서 보는 센서 모듈을 이용하여 수행된다. 센서 모듈은 그림 1(a)에서 볼 수 있듯이 이동 로봇의 전면 부에 위치한다. 그림 1(b)에서 보면 두 개의 laser beam이 존재하여 수평의 sheet of light를 만들어 낸다. 이렇게 만들어진 laser beam이 장애물에 반사가 되면

* 책임저자(Corresponding Author)

논문접수 : 2004. 1. 20., 채택확정 : 2005. 4. 7.

노동기 : LG 필립스(dohn79electronics@hotmail.com)

김곤우, 이범희 : 서울대학교 전기공학부

(kgw0510@snu.ac.kr/bhlee@asri.snu.ac.kr)

vision sensor에 의해 laser beam의 영상이 얻어진다. 이러한 영상에 대해 영상 처리과정을 거쳐 laser beam의 형상을 추출할 수 있다. 이때 laser beam과 vision sensor간의 기하학적 분석을 통해 거리를 측정한다. 또한 상, 하 두 개의 laser beam을 부착함으로써 인해 거리의 신뢰성을 높일 수 있다. 다시 말해 장애물의 3차원적인 정보를 간략하게나마 얻을 수 있어서 거리의 신뢰도를 높일 수 있다는 것이다.

III. Local Map Building

Local map 생성은 laser beam과 vision sensor를 이용하여 수행된다. laser beam을 동적 marker로 사용하여 주변 환경을 인식하게 된다. Vandorpe는 laser beam의 structured light 특징을 이용하여 주변을 인식하였다[6].

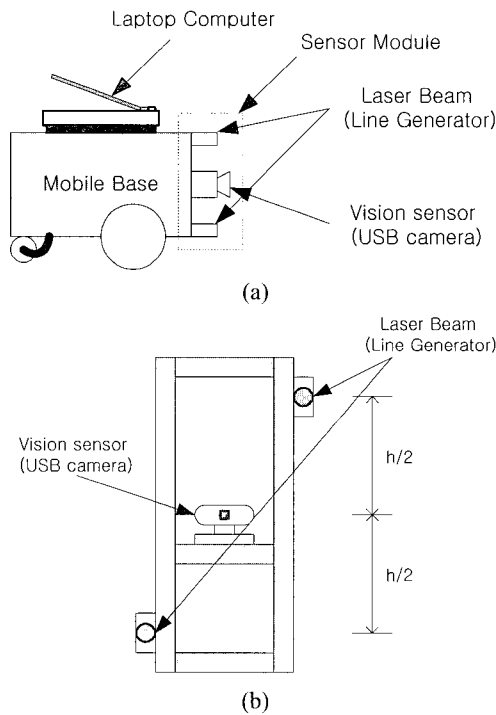


그림 1. (a) 이동 로봇 시스템 (b) 두 개의 laser beam과 vision sensor를 이용한 센서 모듈.

Fig. 1. (a) Mobile robot system (b) sensor module with dual laser beams and vision sensor.

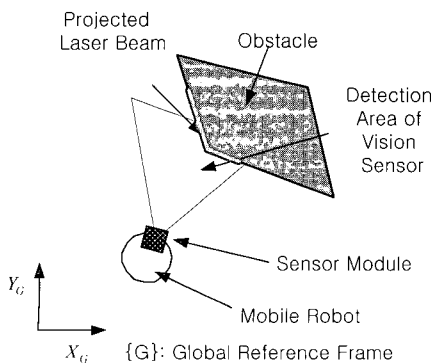


그림 2. 로봇의 물체 인식.
Fig. 2. Recognition of an obstacle.

그림 2와 같이 laser beam을 투사한 후에 카메라를 통해 얻어진 laser beam의 형상을 이용하여 주변 환경을 인식하게 된다.

그림 2와 같이 물체에 laser beam을 투사하게 되면 물체 모양에 따라 투사된 형태가 달라지게 되는데 이때 장애물에 의해 얻어지는 laser beam의 영상은 그림 3(a)와 같다. 이렇게 얻어진 laser beam은 camera와 laser beam간의 기하학적인 분석을 통해 그림 3(b)와 같이 local map을 생성하게 된다. 또한 상하의 laser beam에 대해 각각의 local map이 생성된다.

Local map 생성은 크게 세 부분으로 나눌 수 있다. 첫 번째 부분은 camera calibration을 통해 vision sensor의 intrinsic parameter를 추정한다[7]. 다음으로는 카메라로부터 얻은 영상에 대해 laser beam을 검출하는 부분이다. 이렇게 검출된 영상과 vision sensor의 intrinsic parameter, camera의 geometric 분석을 통하여 영상에서 검출된 laser beam의 pixel 좌표를 local map

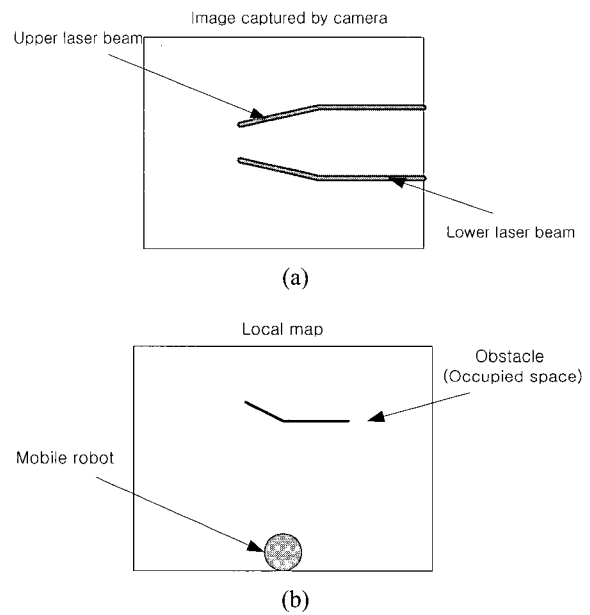


그림 3. (a) Vision sensor에 검출된 laser beam 영상 (b) laser beam 영상에 의해 생성된 local map.

Fig. 3. (a) Image of laser beam detected by a vision sensor (b) local map generated by laser beam.

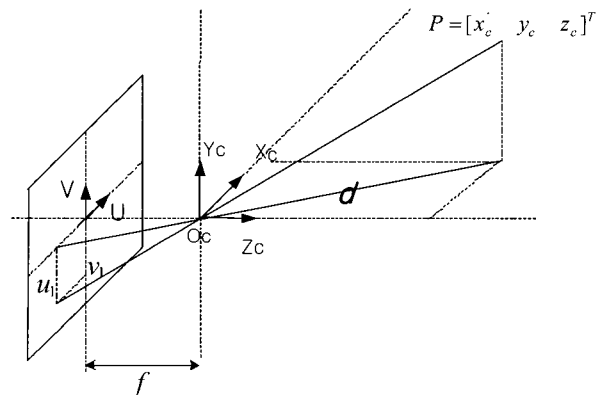


그림 4. 카메라와 laser beam의 geometric 모델.
Fig. 4. Geometric model of camera and laser beam.

좌표로 변환한다. Local map 작성을 위한 좌표 값은 그림 4와 같은 geometric 분석을 통하여 변환된다.

그림 4에서 보는 바와 같이 world coordinate에서 laser beam은 image 평면상의 점 (u_1, v_1) 에 맺히게 된다. 이때 image 평면상의 좌표 값을 실제의 거리 값으로 바꾸기 위한 conversion parameter, k , 를 이용하여 pixel의 위치와 metric space간의 관계를 (1)과 같이 찾아낸다.

$$(u_1, v_1) \Rightarrow [k_u u_1, k_v v_1]^T \leftrightarrow (pixel \Rightarrow mm) \quad (1)$$

k 값을 이용하여 pixel값을 mm단위로 변환한 다음 (2)를 이용하여 검출한 물체까지의 거리 값을 계산하여 local map을 생성한다. (2)에서 y_c 값은 laser beam의 높이를 나타내는 값으로 일정하다.

$$d = \frac{\sqrt{f^2 + (k_u u_1)^2}}{k_v v_1} y_c \quad (2)$$

where $y_c = h / 2$

f 는 camera의 focal length값이고 d 는 camera에서부터 검출한 물체까지 거리이다. u, v 는 각각 image 평면에서 중심에서부터 U축, V축 상의 좌표 값을 나타낸다. 위 식을 이용하여 camera의 image 평면상에 맺힌 laser beam을 world coordinate상의 거리로 변환할 수 있다. Laser beam이 검출된 각각의 pixel에 대해 계산된 거리 값을 이용하여 현재 frame에서 local map을 생성한다. 현재 구한 거리는 camera frame에서의 거리 값이므로 camera frame과 robot frame간의 변환식을 이용하여 로봇과 장애물간의 거리를 알 수 있다.

센서모듈에 의해 측정된 거리를 이용하여 grid 형태의 local map을 작성한다. 먼저 n번째 local map을 W_n 이라 하면 다음과 같이 나타낼 수 있다.

$$W_n = \{(x, y) \text{ in view sector where } C(x, y) = 1\}$$

where $C(x, y) = \begin{cases} 1 & \text{if occupied} \\ 0 & \text{if empty} \end{cases}$

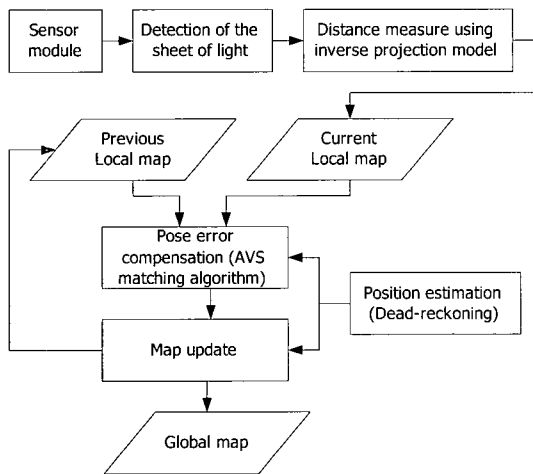


그림 5. 상대적인 위치 추정 및 전체 지도 작성을 위한 전체 시스템의 구조.

Fig. 5. The framework of the overall system for the relative localization and the global map building.

IV. Relative Localization using AVS Matching Algorithm

1. Global Map Building

Global map은 연속적으로 생성되는 local map과 로봇의 위치 정보 등을 이용하여 갱신된다[8]. 이때 중요한 것은 각각의 local map을 이용하여 global map을 갱신하는 방법이다. 기본적으로 연속적으로 생성된 local map과 dead-reckoning에 의한 로봇의 pose 정보를 이용하여 global map을 생성한다. 이때 발생 가능한 문제는 dead-reckoning에 의해 추정되는 로봇의 pose 정보가 정확하지 않다는 것이다. 하지만 laser beam과 vision sensor를 이용하여 생성되는 local map의 정보가 충분히 정확하게 환경을 모델링을 할 수 있으므로 이를 이용하여 로봇의 위치 오차를 보정할 수 있다. 즉, (n+1)번째 생성된 상·하 두 개의 local map과 n번째 생성된 두 개의 local map을 로봇의 pose 정보와의 관계에 따라 map matching과정을 수행하여 위치 오차를 추정, 보완하여 global map을 update하는 방법을 제안한다.

본 논문에서 상대 위치를 추정하고 global map을 생성하는 과정은 크게 세가지 부분으로 나눌 수 있다. 첫 번째로 센서 모듈을 이용하여 local map을 생성한다. 두 번째로 로봇의 odometry data와 local map matching을 통해 로봇의 위치를 추정한다. 세 번째로 추정된 로봇의 위치를 이용하여 local map을 global map에 추가한다. Global map은 상·하 두 개의 local map을 비교하여 최소 거리값을 이용하여 update한다. 이러한 과정에 대한 전체 시스템 구조는 그림 5와 같다.

2. Pose Error Compensation Algorithm

로봇 실제 좌표 추정은 상대적인 위치 추정 방법으로 이는 global map을 생성하는데 중요한 역할을 한다. Global map과 local map을 이용한 로봇 좌표 추정에 관한 연구는 Gonzalez에 의해 제안된바 있다[4]. 본 논문에서는 연속되는 두 개의 local map과 odometry 정보를 이용하여 로봇 좌표를 추정한다. 또한 vision sensor의 검출 영역 특징을 이용한 특정 영역 비교를 통해 실제 좌표를 추정한다.

2.1. AVS (Aligned View Sector) 영역 설정

Vision sensor의 특성상 로봇이 한번에 검출 할 수 영역이 제한된다. 그림 6과 같이 로봇이 이동할 때 이동 거리가 크지 않다면 로봇은 global 좌표상에서 동일한 영역을 중복하여 검출하게 된다. 이렇게 중복하여 검출되는 영역을 AVS (Aligned View Sector) 영역이라고 정의한다.

Sensor module은 일정한 sample time마다 sensor data를 읽어오게 된다. 이때 n번째 sensor reading후 로봇이 이동하고 (n+1)번째 sensor reading을 하게 되면 로봇은 그림 6과 같이 중복된 영역을 검출하게 된다. 이때 중복하여 검출된 물체는 global reference frame에 대해 동일한 좌표 값을 가져야 한다. 하지만 dead-reckoning 정보의 부정확성으로 인해 중복된 영역에서의 좌표 값에 차이가 발생하게 된다. 이러한 차이를 이용하여 상대적인 위치 추정(relative localization)을 하게 된다. 중복된 영역의 matching을 이용하여 상대적인 위치 추정(relative localization)을 위해서는 먼저 AVS 영역을 찾아야 한다.

기본적으로 이동 로봇의 위치 표시를 위해 그림 7(a)와 같이 좌표계를 정의한다. 먼저 전체 지도에 대한 좌표계로 global reference frame, {G},를 정의하고 global reference frame 안에서 로봇의 좌표계로 robot local reference frame, {R}, 을 정의

한다. Global reference frame에서의 이동 로봇의 위치를 표시하기 위해서는 configuration space (C-space)상의 로봇 정보를 나타내야 하며 이는 그림 7(b)와 같이 위치와 각도 정보를 이용하여 로봇의 위치(robot pose)를 나타낸다.

그림 7(b)에 표시된 것과 같이 n번째 로봇 위치는 C-space 상에서 x_n, y_n, θ_n 으로 표시된다. θ_n 는 vision sensor가 갖는 시야각이고 d_{max} 는 최대 검출 거리를 나타낸다. Vision sensor가 갖는 시야각은 camera의 field of view angel에 의해 결정된다. 또한 최대 검출 거리는 sensor module의 resolution을 감안하여 임의로 결정할 수 있다. 다음으로 검출 범위의 경계를 구하기 위하여 좌우 측의 경계에 대한 view line을 다음과 같이 정의한다.

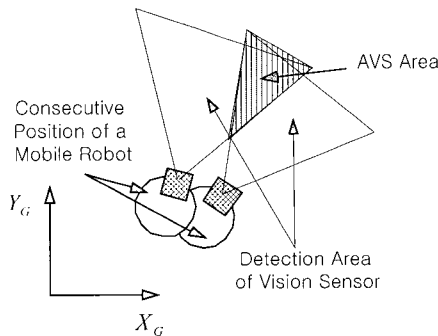


그림 6. Vision sensor의 검출 영역 및 AVS 영역.
Fig. 6. Detection area of a vision sensor and AVS area.

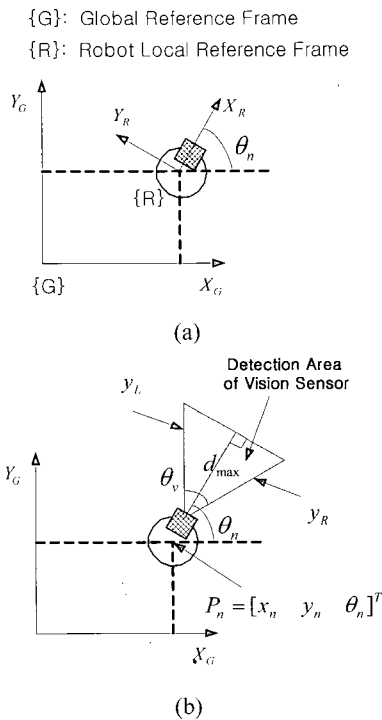


그림 7. (a) Global reference frame과 로봇의 local reference frame
(b) configuration space (C-space)상에서의 로봇의 위치 정보와 vision sensor의 검출 영역.

Fig. 7. (a) Global reference frame and robot local reference frame
(b) robot pose data in C-space and detection area of a vision sensor.

$$y_R = \tan\left(\theta_n - \frac{\theta_v}{2}\right)(x - x_n) + y_n \quad (3)$$

$$y_L = \tan\left(\theta_n + \frac{\theta_v}{2}\right)(x - x_n) + y_n \quad (4)$$

Global reference frame에서 AVS 영역을 계산하고 영역 내에서의 map matching을 수행하기 위해서는 많은 기하학적 계산들이 요구된다. 또한 본 논문에서 제안하는 알고리즘은 두 개의 연속된 local map에 대한 AVS 영역 계산 및 map matching을 수행하는 것이다. 따라서 n번째 로봇에 대한 robot local reference frame을 기준으로 AVS 영역을 구하고 map matching을 수행한다. 이때 global reference frame과 robot local reference frame 간의 좌표 변환 식은 다음과 같다.

$$\begin{bmatrix} x_g \\ y_g \\ \theta_g \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta_n & -\sin \theta_n & 0 & x_n \\ \sin \theta_n & \cos \theta_n & 0 & y_n \\ 0 & 0 & 1 & \theta_n \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r \\ y_r \\ \theta_r \\ 1 \end{bmatrix} \quad (5)$$

로봇의 local reference frame에서의 로봇 위치를 $[x_r, y_r, \theta_r]^T$ 라 하고 global reference frame에서의 n번째 로봇의 위치를 $[x_n, y_n, \theta_n]^T$ 이라 했을 때 로봇의 local reference frame에서의 로봇 위치는 global reference frame에서 $[x_g, y_g, \theta_g]^T$ 와 같이 변환된다. 따라서 n번째 로봇에 대한 local reference frame을 기준으로 하여 (n+1)번째 로봇의 상대적인 위치를 구하고 이에 대한 AVS 영역을 구한다. 이 AVS (Aligned View Sector) 영역은 로봇의 이동 정보와 그림 7(b)에서의 로봇의 검출영역에 관한 정보를 이용하여 계산하게 된다. AVS 영역은 로봇의 이동 정보에 따라 달라지며 이동 정보는 다음과 같이 2가지 경우로 나누어 볼 수 있다. 먼저 로봇의 translational motion의 경우와 로봇의 rotational motion의 경우로 나눌 수 있다. 이때 각각의 motion은 독립적으로 발생하는 것을 가정한다.

그림 8에서 보는 것과 같이 n 번째 로봇의 local reference frame을 기준으로 했을 때의 좌표계를 구성할 수 있다. Global reference frame에서의 (n+1)번째 로봇의 위치 정보는 n번째 로봇의 위치 정보와 odometry 정보를 이용하여 다음과 같이 구할 수 있다.

$$P_{n+1} = \begin{bmatrix} x_{n+1} \\ y_{n+1} \\ \theta_{n+1} \end{bmatrix} = \begin{bmatrix} x_n \\ y_n \\ \theta_n \end{bmatrix} + \begin{bmatrix} \delta x \\ \delta y \\ \delta \theta \end{bmatrix} \quad (6)$$

그림 8의 local reference frame에 대한 (n+1)번째 로봇의 위치는 $\delta x, \delta y, \delta \theta$ 에 의해 결정된다. 이때 $\delta x, \delta y$ 는 로봇의 translational motion에 대한 변위를 나타내며 $\delta \theta$ 는 로봇의 rotational motion에 대한 변위를 나타낸다. 따라서 각각의 변위에 대한 AVS 영역을 설정한다. 이때 AVS 영역은 3개의 점을 잇는 삼각형의 형태로 설정한다. 그림 8에서 보는 바와 같이 vision sensor의 검출 영역이 결정되고 이때 view line은 다음과 같이 정의할 수 있다.

$$y_R = \tan\left(-\frac{\theta_v}{2}\right)x \quad (7)$$

$$y_L = \tan\left(\frac{\theta_v}{2}\right)x \quad (8)$$

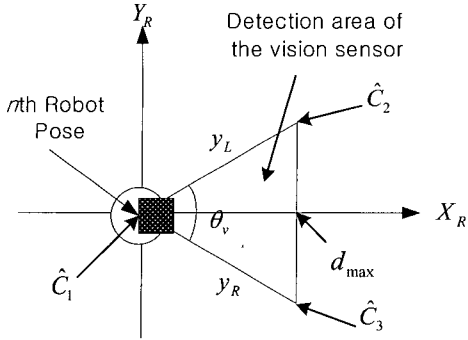


그림 8. n번째 로봇 위치에서의 local reference frame.

Fig. 8. Robot local reference frame for nth robot pose.

그리고 vision sensor의 검출 영역에 대한 세 점을 다음과 같이 구할 수 있다.

$$\hat{C}_1 = [0 \ 0]^T \tag{9}$$

$$\hat{C}_2 = \left[d_{\max} \quad d_{\max} \cdot \tan\left(\frac{\theta_v}{2}\right) \right]^T \tag{10}$$

$$\hat{C}_3 = \left[d_{\max} \quad d_{\max} \cdot \tan\left(-\frac{\theta_v}{2}\right) \right]^T \tag{11}$$

2.1.1 Translational Motion에 대한 AVS 영역 설정

1) X_R 축을 따라 motion이 발생한 경우

X_R 축에 대한 로봇의 translational motion, δx , 가 발생한 경우 그림 9(a)의 빗금친 부분과 같이 AVS 영역을 설정할 수 있다. 이때 δx 가 양수일 때와 음수일 때 각기 다른 부분에서 AVS 영역이 형성되는 것을 알 수 있다. 따라서 각각의 경우에 대해 AVS 영역을 구하고 이를 signum function을 이용하여 다음과 같은 AVS 영역을 설정할 수 있다.

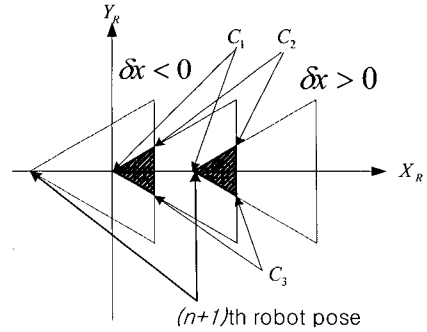
$$C_1 = \hat{C}_1 + \begin{bmatrix} \frac{\delta x}{2} (\text{sgn}(\delta x) + 1) \\ 0 \end{bmatrix} \tag{12}$$

$$C_2 = \hat{C}_2 + \begin{bmatrix} \frac{\delta x}{2} (\text{sgn}(-\delta x) + 1) \\ \delta x \cdot \text{sgn}(-\delta x) \cdot \tan\left(\frac{\theta_v}{2}\right) \end{bmatrix} \tag{13}$$

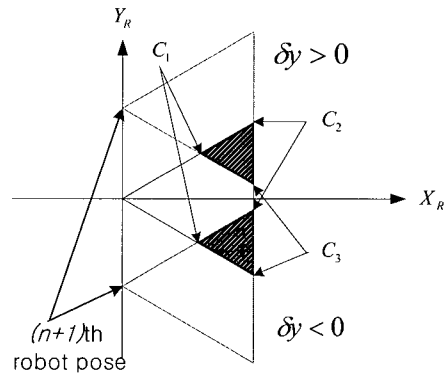
$$C_3 = \hat{C}_3 + \begin{bmatrix} \frac{\delta x}{2} (\text{sgn}(-\delta x) + 1) \\ \delta x \cdot \text{sgn}(-\delta x) \cdot \tan\left(-\frac{\theta_v}{2}\right) \end{bmatrix} \tag{14}$$

2) Y_R 축을 따라 motion이 발생한 경우

Y_R 축에 대한 로봇의 translational motion, δy , 가 발생한 경우 그림 9(b)의 빗금친 부분과 같이 AVS 영역을 설정할 수 있다. 엄밀히 말해서 omni-directional robot의 경우가 아니면 nonholonomic constraint에 따라 Y_R 축을 따라 이동할 수는 없다. 하지만 omni-directional robot의 확장성과 Y_R 축을 따라서 발생할 수 있는 오차를 고려한다면 Y_R 축을 따라 발생하는 motion에 대한 분석이 필요하다. Y_R 축에 대해서도 δy 가 양수일 때와 음수일 때 각기 다른 부분에서 AVS 영역이 형성



(a) AVS area for δx



(b) AVS area for δy

그림 9. Translational motion에 대한 AVS영역.

Fig. 9. AVS for translational motion of robot.

된다. 따라서 각각의 경우에 대해 AVS 영역을 구하고 이를 통합하여 다음과 같은 AVS 영역을 설정할 수 있다.

$$C_1 = \hat{C}_1 + \begin{bmatrix} \frac{\delta y}{2} \text{sgn}(\delta y) \cdot \cot\left(\frac{\theta_v}{2}\right) \\ \frac{\delta y}{2} \end{bmatrix} \tag{15}$$

$$C_2 = \hat{C}_2 + \begin{bmatrix} 0 \\ \frac{\delta y}{2} (\text{sgn}(\delta y) + 1) \end{bmatrix} \tag{16}$$

$$C_3 = \hat{C}_3 + \begin{bmatrix} 0 \\ \frac{\delta y}{2} (\text{sgn}(\delta y) + 1) \end{bmatrix} \tag{17}$$

2.1.2 Rotational Motion에 대한 AVS 영역 설정

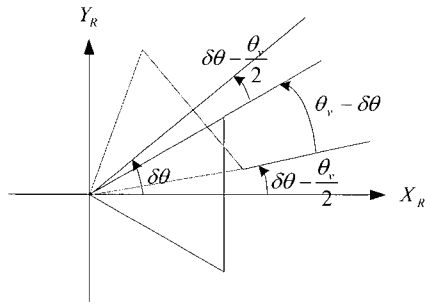
Rotational motion에 대한 AVS 영역은 그림 10과 같다. 2.1.1 절과 같은 방법으로 $\delta\theta$ 가 양수일 때와 음수일 때 각각 형성되는 AVS 영역을 다음과 같이 설정할 수 있다.

$$C_1 = \hat{C}_1 \tag{18}$$

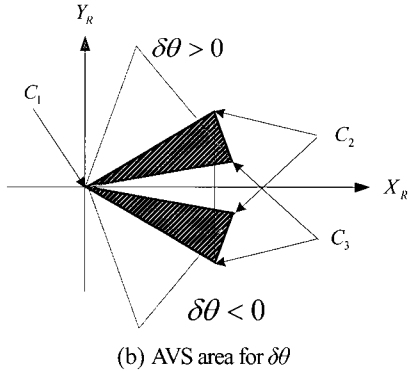
$$C_2 = R\left(\frac{\delta\theta}{2} (\text{sgn}(-\delta\theta) + 1)\right) \hat{C}_2 \tag{19}$$

$$C_3 = R\left(\frac{\delta\theta}{2} (\text{sgn}(\delta\theta) + 1)\right) \hat{C}_3 \tag{20}$$

where $R(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$



(a) The geometric characteristic for rotational motion



(b) AVS area for $\delta\theta$

그림 10. Rotational motion에 대한 AVS 영역.
Fig. 10. AVS for rotational motion of robot.

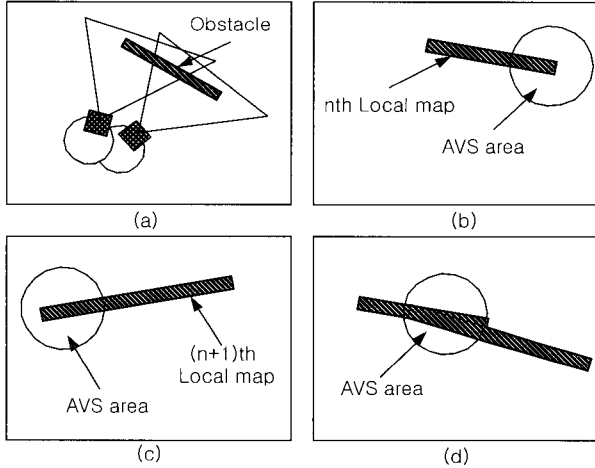


그림 11. (a) 로봇의 이동에 따른 검출 영역 (b) n번째 로봇 위치에서의 local map (c) (n+1)번째 로봇위치에서의 local map (d) 두 개의 연속된 local map에 의해 만들어진 global map.

Fig. 11. (a) View sector for motion of robot (b) local map at nth robot position (c) local map at (n+1)th robot position (d) global map combined with two consecutive local maps.

2.2. Pose Error Compensation

검출 영역이 설정되면 그 영역을 이용하여 로봇의 실제 좌표를 추정하게 된다. 이와 관련하여 Gonzalez는 연속하여 생성된 local map을 이용하여 로봇의 위치를 추정하는 방법을 제안하였다[9]. 본 논문에서는 이동하지 않는 고정 장애물

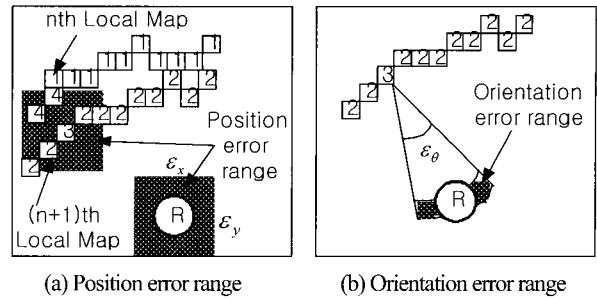


그림 12. 로봇의 오차 범위.
Fig. 12. Error ranges of the robot.

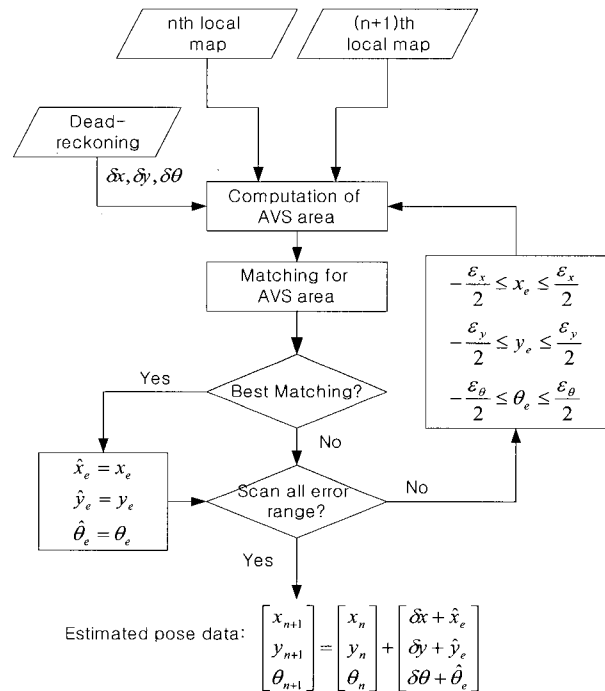


그림 13. 위치 오차 보정 알고리즘 수행 순서도.
Fig. 13. Block diagram of pose error compensation algorithm.

을 서로 다른 위치에서 검출하였을 때 global 좌표상에서 같은 좌표 값을 가진다는 조건을 이용하여 실제 로봇 좌표 값을 추정하는 알고리즘을 제안한다. 이를 위해서 다음과 같이 세가지 가정을 한다.

가정 1: 주변 환경은 모두 고정 장애물이다.

가정 2: 일정한 이동량에 대해 발생하는 오차는 일정 범위 내에 존재한다.

그림 11에서 보는 것과 같이 서로 다른 위치에서 검출한 장애물은 동일한 위치에 있음에도 불구하고 로봇의 dead-reckoning에 의한 위치 정보의 오차 때문에 global 좌표상에서 두 개의 벽면의 좌표가 일치하지 않는 경우가 발생한다. 다시 말해 이러한 문제는 생성된 각각의 local map을 global map에 추가하는 과정에서 global 좌표상의 로봇 좌표의 부정확성으로 인해 발생된다. 따라서 본 논문에서는 그림 11(d)에서와 같이 AVS 영역에서 두 개의 local map의 matching 정도를 이용하여 지도의 틀어진 정보를 측정하고 이를 이용하여 로봇의 실제 좌표를 추정하게 된다. 여기서 상하 두 개의 laser

beam에 의해 생성된 두 개의 local map에 대해 각각 matching을 하여 위치 오차 추정치의 신뢰성을 향상시킨다.

연속되는 로봇의 위치간에 발생하는 오차는 기구적인 요인과 비기구적인 요인에 의해 발생하며 이는 일정한 범위 안에서 존재한다. 이러한 오차 범위는 그림 12와 같이 표현할 수 있다. 본 논문에서는 위치 오차(position error)와 방위 오차(orientation error)의 한계 값을 설정하고 그 한계 값 내에서 실제 로봇 좌표와 가장 가까운 좌표 값을 추정한다. 즉, 로봇의 dead-reckoning 방법에 의해 발생할 수 있는 오차범위 안에서 (n+1)번째 로봇의 위치가 변화하였을 때 local map이 n번째 로봇의 위치에서의 local map과 가장 정확하게 matching이 되는 오차 값을 측정할 수 있다. 이때 AVS 영역에 포함되는 영역에 대해서 matching을 수행한다. 이러한 좌표 보정 알고리즘의 수행 과정은 그림 13과 같다.

그림 12(a),(b)는 로봇의 오차 범위를 나타내고 있다. 그림 12은 n번째 local map과 (n+1)번째 local map에 대한 AVS영역을 나타내고 있다. 이때 그림 12(a)와 (b)는 각각 위치 오차 범위와 방위 오차 범위를 나타내고 있다.

그림 12에서 '2'와 '3'으로 표시된 grid는 (n+1)번째 로봇 위치에서 작성한 local map에 해당하는 것이며 '1'과 '4'로 표시된 grid는 n번째 로봇 위치에서 작성된 local map이다. 이때 n번째 로봇 위치와 (n+1)번째 로봇 위치에서 작성된 local map은 동일한 물체를 검출한 것이다. 그런데 두 지도가 정확하게 겹치지 않음을 볼 수 있는데 이는 이동한 로봇 좌표가 정확하지 않기 때문이다. 가정 2에서 오차 범위가 일정하다고 가정하였으므로 그림 12의 빗금 친 영역과 같이 오차 범위를 구할 수 있다.

이때 각각의 오차 범위를 ϵ_x , ϵ_y , ϵ_θ 라고 정의한다. 따라서 오차 범위 안에서 오차가 존재할 때 적절한 AVS영역을 찾아야 한다. 이때 로봇의 오차 범위는 로봇의 이동 정보와 비교해서 상대적으로 작은 값을 갖는다. 여기서 로봇의 이동은 크게 두 가지로 나누어 고려한다. 첫째는 translational motion이고 다른 하나는 rotational motion이다. 다음으로 각각의 motion에 대하여 오차 범위내의 추정 오차, (x_e, y_e, θ_e) , 에 대한 AVS 영역 갱신하고 갱신된 AVS 영역에 대한 local map matching을 통하여 가장 잘 matching될 때의 오차 값을 찾아내어 로봇의 odometry data를 보정해 준다.

2.2.1 AVS 영역 갱신

그림 14(a)는 translational motion에 대한 AVS 영역의 변형을 보여주고 있다. 먼저 이동 로봇이 omni-directional motion을 하지 않는다고 가정했을 때 translational motion은 X_R 축을 따라 로봇이 δx 만큼 이동하는 것을 의미한다. 여기서 X_R 축에 대한 추정 오차, x_e 는 δx 에 비해 상대적으로 작은 값을 갖는다. 따라서 AVS 영역은 (12), (13), (14)를 이용하여 구할 수 있다.

$$C_1 = \hat{C}_1 + \begin{bmatrix} \frac{(\delta x + x_e)}{2} (\text{sgn}(\delta x) + 1) \\ y_e \end{bmatrix} \quad (21)$$

$$C_2 = \hat{C}_2 + R(\theta_e) \cdot \begin{bmatrix} \frac{(\delta x + x_e)}{2} (\text{sgn}(-\delta x) + 1) \\ (\delta x + x_e) \cdot \text{sgn}(\delta x) \cdot \tan\left(\frac{\theta_e}{2}\right) + y_e \end{bmatrix} \quad (22)$$

$$C_3 = \hat{C}_3 + R(\theta_e) \begin{bmatrix} \frac{(\delta x + x_e)}{2} (\text{sgn}(-\delta x) + 1) \\ (\delta x + x_e) \cdot \text{sgn}(-\delta x) \cdot \tan\left(-\frac{\theta_e}{2}\right) + y_e \end{bmatrix} \quad (23)$$

다음으로는 rotational motion에 대한 AVS영역을 갱신하는 방법은 그림 14(b)에 나타나 있다. 로봇의 rotational motion에 대한 추정 오차, θ_e 는 $\delta\theta$ 에 비해 상대적으로 작은 값을 갖는다. 따라서 (18), (19), (20)을 이용하여 다음과 같이 AVS 영역을 구할 수 있다.

$$C_1 = \hat{C}_1 + \begin{bmatrix} x_e \\ y_e \end{bmatrix} \quad (24)$$

$$C_2 = R\left(\frac{(\delta\theta + \theta_e)}{2} (\text{sgn}(-\delta\theta) + 1)\right) \hat{C}_2 + \begin{bmatrix} x_e \\ y_e \end{bmatrix} \quad (25)$$

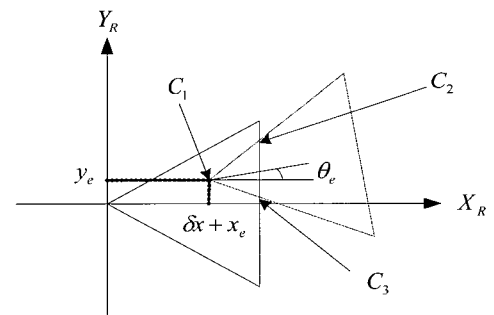
$$C_3 = R\left(\frac{(\delta\theta + \theta_e)}{2} (\text{sgn}(\delta\theta) + 1)\right) \hat{C}_3 + \begin{bmatrix} x_e \\ y_e \end{bmatrix} \quad (26)$$

2.2.2 로봇 좌표 보정

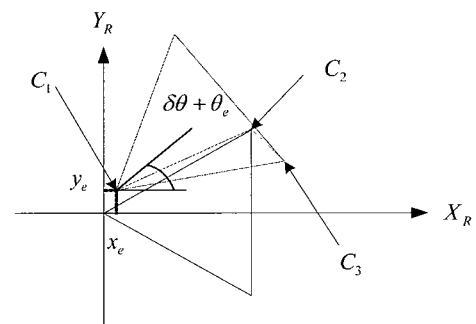
로봇의 위치 오차를 보정하기 위해서 생성된 local map의 wall정보만을 이용한다. 일반적인 위치 추정 알고리즘은 생성된 local map의 전 영역을 이용하여 실제 위치를 추정하는 반면 본 논문에서는 AVS영역에서 생성된 wall 정보만을 이용하게 된다.

먼저 global map을 G 라 하면 다음과 같이 나타낼 수 있다.

$$G = \bigcup_{n=1}^{\infty} W_n \quad (27)$$



(a) AVS area update for translational motion



(b) AVS area update for rotational motion

그림 14. AVS 영역 갱신.

Fig. 14. AVS area updates.

이때 n번째 local map, W_n 은 로봇의 위치 정보, (x_n, y_n, θ_n) , 에 의해 좌우된다. 여기서 n번째 local map과 (n+1)번째 local map에 대한 AVS 영역은 다음과 같다.

$$W_{n,n+1}^{AVS} = W_n \cap W_{n+1} \quad (28)$$

이때 n번째 local map에 대한 로봇의 위치를 기준으로 (n+1)번째 로봇의 위치가 오차 범위내에서 변하기 때문에 AVS영역 또한 변한다.

여기서 AVS 영역에 대해 n번째 local map과 (n+1)번째 local map의 matching 정도를 알아보기 위한 matching score를 다음과 같이 정의할 수 있다.

$$Matching\ Score = \sum_{(x,y) \in W_{n,n+1}^{AVS}} c_n(x,y) \cdot c_{n+1}(x,y) \quad (29)$$

where $c_k(x,y) = C(x,y)$ for $(x,y) \in W_k$

$$C(x,y) = \begin{cases} 1 & \text{if occupied} \\ 0 & \text{if empty} \end{cases}$$

위의 matching score를 이용하면 두 개의 연속된 map에 대한 일치 정도를 정량적으로 구할 수 있다. 여기서 n번째 로봇의 위치를 기준으로 (n+1)번째 로봇의 위치를 추정하는데 위의 matching score를 이용한다.

먼저 (n+1)번째 로봇의 위치는 odometry 정보를 기준으로 하여 로봇의 위치 오차 범위내의 오차를 갖는다. 이때 오차 범위내에서 (n+1)번째 로봇의 위치 좌표를 바꾸어 가며 그때의 local map과 n 번째 local map이 가장 잘 일치할 때의 추정 오차, (x_e, y_e, θ_e) , 를 찾는다. 즉, (29)에서 matching score 가장 크게 만드는 추정 오차 값, $[\hat{x}_e \ \hat{y}_e \ \hat{\theta}_e]^T$, 을 이용하여 로봇 좌표 값을 다음과 같이 보정한다.

$$\begin{bmatrix} \hat{x}_{n+1} \\ \hat{y}_{n+1} \\ \hat{\theta}_{n+1} \end{bmatrix} = \begin{bmatrix} x_n \\ y_n \\ \theta_n \end{bmatrix} + \begin{bmatrix} \delta x + \hat{x}_e \\ \delta y + \hat{y}_e \\ \delta \theta + \hat{\theta}_e \end{bmatrix} \quad (30)$$

where $\begin{bmatrix} \hat{x}_e \\ \hat{y}_e \\ \hat{\theta}_e \end{bmatrix} = \arg \max \text{ Matching Score}$

Matching score의 신뢰성 향상을 위하여 상·하의 laser beam에 대해 생성된 local map에 대하여 각각 matching score를 계산하여 upper local map과 lower local map의 matching score의 합을 이용한다.

V. 실험 결과

1. 실험 시스템 구성

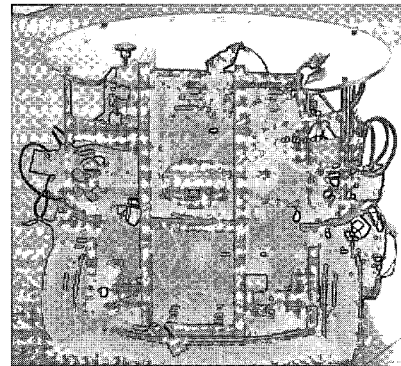
실험 시스템의 구성은 그림 15(a)와 같이 이동 로봇에 센서 모듈을 부착하고 있다. 센서 모듈은 그림 15(b)와 같이 두 개의 laser beam과 CCD카메라로 구성되어 있다. 실험에 사용된 laser beam은 LANICS사의 LM-6305ML 제품으로 fan angle이 60°이고 출력은 2.5mW인 laser line generator이다. 이러한 line generator를 지면과 수평하게 투사하도록 센서 모듈을 제작하였다. 그리고 CCD 카메라는 Intel CS330 USB 카메라를 사용하고 있으며 카메라의 resolution은 640x480이며 Field of view는 50° (<5% distortion) 이다.

센서 모듈을 이용하여 측정할 수 있는 최대거리는 대략 1.5m정도이며 이는 laser beam과 카메라 간의 거리에 따라 달라질 수 있다. 이때 측정된 거리정보를 이용하여 생성하는 map은 1cm x 1cm grid를 갖는 grid map 방식을 이용한다.

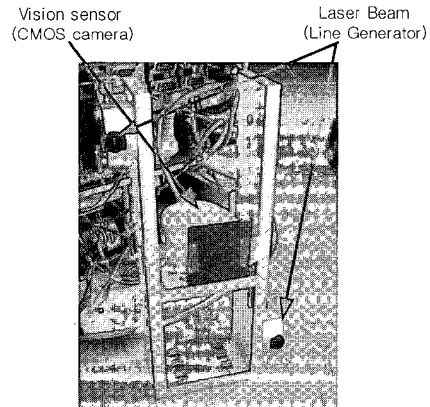
2. 실험 결과

로봇의 odometry error는 Gaussian model을 이용하여 모델링 하였으며 따라서 오차 범위는 이동 거리에 비례한다.

먼저 첫번째 실험에서는 rotational motion에 대하여 오차 보정 및 global map 작성 실험을 수행하였다. 실험 환경은 그림 16과 같이 구조화된 환경에서 실험을 수행하였으며 로봇을 제자리에서 20° 만큼 4회를 회전하며 각각의 local map을 작성하였다.



(a)



(b)

그림 15. (a) 이동 로봇 시스템 (b) 센서 모듈.

Fig. 15. (a) Mobile robot system (b) sensor module.

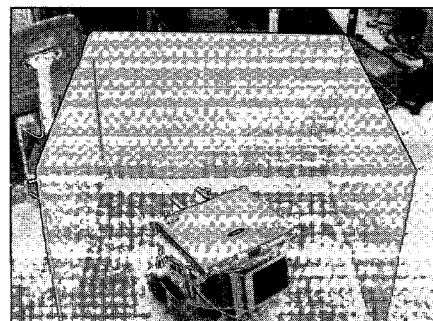


그림 16. 회전 이동에 대한 실험 환경.

Fig. 16. Experimental environment for rotational motion.

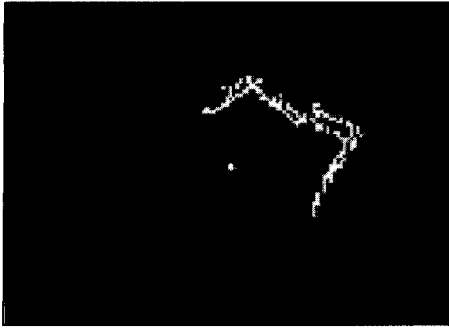


그림 17. 보정 전에 생성된 global map.
Fig. 17. Global map without error correction.

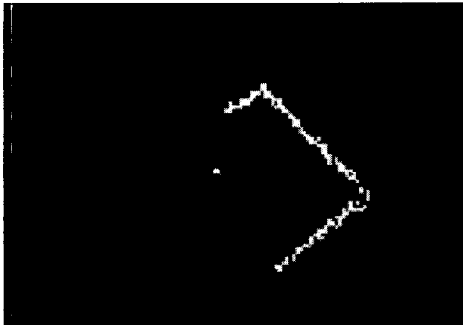


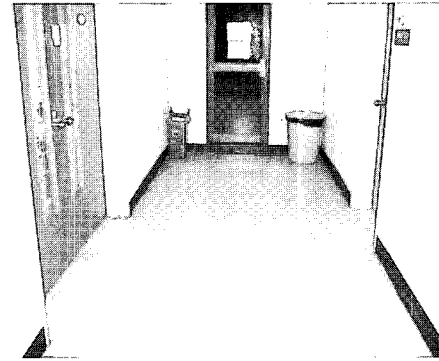
그림 18. 보정 후에 생성된 global map.
Fig. 18. Global map with error correction.

그림 17은 상대 위치 추정 알고리즘을 적용하지 않은 결과를 보여주고 있다. 로봇의 odometry 정보의 오차로 인하여 센서 모듈을 이용하여 작성된 local map이 global map에 정확하게 추가되지 않음을 볼 수 있다. 반면에 그림 18은 상대 위치 추정 알고리즘을 적용하여 좀 더 개선된 결과를 보여주고 있다.

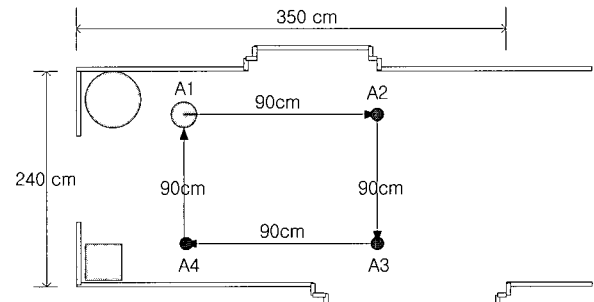
다음으로 실제 환경에 적용하여 성능을 확인하기 위한 실험을 수행하였다. 실험을 위해 복도 환경에서 로봇의 translational motion과 rotational motion을 동시에 수행하며 각각의 위치에서 생성된 local map을 이용하여 로봇의 상대 위치 추정과 global map 작성을 하는 실험을 수행하였다.

실험 환경은 그림 19와 같다. 먼저 그림 19(a)는 실험을 위한 실내 복도 환경을 보여주고 있다. 실험은 그림 19(b)와 같이 A1→A2→A3→A4→A1으로 이동하며 수행하였다. A1, A2, A3, A4의 위치에서는 20°씩 회전하며 local map을 작성하도록 하였다. 또한 각 위치에서 다른 위치로 이동시 90cm만큼 이동하였다. 이때 실험 결과는 그림 20과 같다. 그림 20(a)는 AVS matching을 통한 상대 위치 추정 알고리즘을 적용하지 않은 결과를 보여준다. 결과에서 보듯이 로봇의 odometry 정보의 오차가 누적되면서 global map의 형태가 많이 어긋나 있음을 알 수 있다. 그러나 AVS matching을 통한 상대 위치 추정 알고리즘을 적용하면 이러한 odometry 정보의 오차를 상당부분 보상할 수 있고 이러한 결과는 그림 20(b)에서 보는 바와 같이 좋은 성능을 나타냄을 알 수 있다.

위와 같이 본 논문에서 제시하는 AVS matching을 이용한 상대 위치 추정 알고리즘의 성능을 검증하였다. 본 알고리즘을 적용하기 위해서는 주변 환경에 대해 local map을 구성할



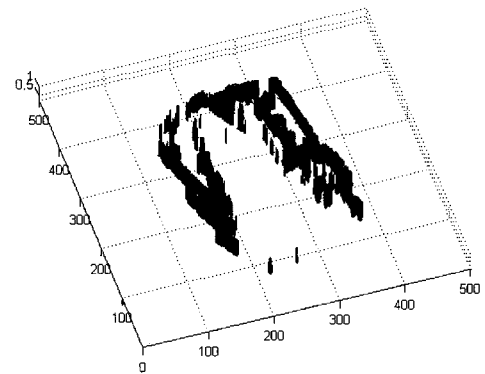
(a) Indoor hallway



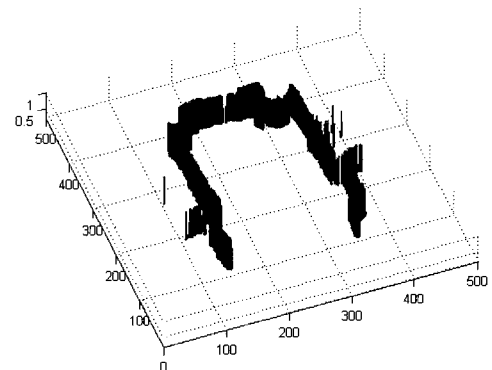
(b) Predefined path for the mobile robot

그림 19. 실험 환경.

Fig. 19. Experimental environment.



(a) Without relative localization using AVS matching



(b) With relative localization using AVS matching

그림 20. 실험 결과.

Fig. 20. Experimental results.

수 있는 장애물이 다수 존재하여야 한다. 또한 오랜 주행으로 누적되는 오차가 존재하며 globally bound되지 않는다. 다만 odometry error로 발생하는 많은 부분의 오차를 AVS matching을 통해 줄일 수 있다.

VI. 결론

본 논문에서는 로봇이 global map 생성을 수행할 때 발생하는 좌표 오차를 최소화 하는 기법을 제안하였다.

로봇에 사용한 센서는 CCD sensor를 기반으로 하는 camera와 structured light인 laser beam을 사용하여 물체를 검출하였다. 검출하는 방법은 laser beam을 물체에 비춘 뒤 laser beam이 변형되는 특징을 이용하여 물체의 형태를 검출하는 방법을 사용하였다. Laser beam과 vision sensor를 이용한 센서 모듈은 초음파 센서와 같은 기존의 sensor에 비해 상대적으로 정확한 거리 측정 결과를 보여주는 것을 알 수 있었다. 이 거리 측정의 정확성을 이용하여 특정 영역에 대한 matching 함수를 수행할 수 있다. 본 논문에서는 laser beam과 vision sensor의 geometric 특성을 이용하여 matching 할 수 있는 영역을 정의하고 그 영역에 대해 matching 알고리즘을 수행하는 방법을 제안하였다.

이러한 방법을 통해 각각의 view sector에서 검출된 동일한 물체를 이용하여 로봇의 좌표를 보정하는 방법을 제안하였다.

참고문헌

- [1] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *IEEE Computer*, vol. 22, pp. 46-57, Jun., 1989.
- [2] J. Borenstein and Y. Koren, "Histogrammic in-motion mapping

for mobile robot obstacle avoidance," *IEEE Trans. on Robotics and Automation*, vol. 7, no. 4, pp. 535-539, Aug., 1991.

- [3] A. Lallement, A. Siadat, M. Dufaut and R. Husson, "Laser-vision cooperation for map building and landmarks recognition," *Proc. of the 1998 IEEE ISIC/CIRA/ISAS Joint Conference*, pp. 387-392, 1998.
- [4] J. Gonzalez, A. Ollero, A. Reina, "Map building for a mobile robot equipped with a 2D laser rangefinder," *Proc. of the 1994 IEEE Int. Conf. on Robotics and Automation*, pp. 1904-1909, 1994.
- [5] H. P. Moravec, "Sensor fusion in certainty grids for mobile robots," *AI Magazine*, vol. 9, no. 2, pp. 61-74, 1988.
- [6] J. Vandorpe, H. Van Brussel and H. Xu, "Exact dynamic map building for a mobile robot using geometrical primitives produced by a 2D range finder," *Proc. of the 1996 IEEE Int. Conf. on Robotics and Automation*, pp. 22-28, 1996.
- [7] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," *Proc. of the Seventh IEEE Int. Conf. on Computer Vision*, pp. 666-673, 1999.
- [8] J. H. Lim and D. W. Cho, "A new certainty grid based mapping and navigation system for an autonomous mobile robot," *Int. Jour. of Advanced Manufacturing Technology*, vol. 10, pp. 139-148, 1995.
- [9] J. Gonzalez, A. Stentz and A. Ollero, "An iconic position estimator for a 2D laser rangefinder," *Proc. of the 1992 IEEE Int. Conf. on Robotics and Automation*, pp. 2646-2651, 1992.
- [10] R. H. T. Chan and P. K. S. Tam, "A new hough transform based position estimation algorithm," *Proc. of the 1994 Second Australian and New Zealand Conf. on Intelligent Information Systems*, pp. 140-144, 1994.



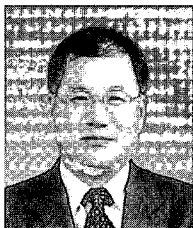
노 동 기

1979년 7월 12일생. 2002년 고려대학교 전기 전자 전파공학부 졸업. 2004년 서울대학교 전기공학부(공학석사). 현재 LG필립스 근무. 관심분야는 global map building.



김 곤 우

1974년 5월 10일생. 2000년 중앙대학교 전기 전자 제어계측공학부 졸업. 2002년 서울대학교 대학원 전기공학부(공학석사). 2002년~현재 서울대학교 대학원 전기공학부 박사과정 재학중. 관심분야는 visual servoing, map building, localization.



이 범 회

1955년 5월 29일생. 1978년 서울대학교 전자공학과(공학사). 1980년 서울대학교 전자공학과(공학석사). 1985년 Univ. of Michigan. Computer, Information & Control Eng. 졸업(공학 박사). 1987년~현재 서울대학교 전기공학부 교수. 관심분야는

로봇 공학.