

스캔코드법을 이용한 이동로봇의 장애물 회피

조 규 상  
 동아대학교 컴퓨터공학부

Obstacle Avoidance of Mobile Robot  
 using Scan Code Method

GyuSang Cho  
 Dept. of Computer Eng., DongYang Univ.

**Abstract** - This paper proposes a scan code method for obstacle avoidance of mobile robot. Obstacles detected in a circular window are converted to scan codes and then to the steering angle. The safe rotating radius is obtained by the scan code to avoid the collision between robot and obstacle and the minimum distance for rotation is calculated. Effectiveness of the method is illustrated through simulations, and the results show that the proposed method can be efficiently implemented to an unknown environment.

1. 서 론

이동 로봇의 장애물 회피를 위한 여러 가지 방법들이 제안되었다. 건물안의 복도와 같은 환경에서 쉽게 적용할 수 있는 방법으로, 미리 정해진 각도로 벽면을 따라 주행하는 WFM(Wall Following Method)[1]방법이 있다. PFM(Potential Field Method)[2]은 장애물이 로봇에 척력을 발휘하고 목표 지점은 인력을 발휘한다고 생각하는 방법이다. 이 방법은 로봇에 가까운 장애물에 대해서 쉽게 적용할 수 있고, 매우 효과적인 방법이다. 이 방법의 최대의 단점은 지역최소점에 빠지기 쉽다는 것이다.

Koren과 Borenstein은 Virtual Force Field Histogram 방법[3]과 이것을 개선한 Vector Field Histogram 방법[4]을 제안하였다. 이 방법에서는 로봇의 환경을 모델링하는데 Occupancy Grid 방법을 사용하여 돌발성 잡음에 대처할 수 있도록 하였고, 격자 정보를 이용하여 로봇 이동 방향과 속도를 계산하는데 사용한다.

최근에 Fox등은[5] 고속으로 장애물이 밀집된 환경에서도 안전하게 제어될 수 있는 Dynamic Window 방법을 이용하여 이동로봇의 충돌회피방법을 제안하였고, Li와 Bui[6]등은 기지의 환경에 적용할 수 있는 유체모델을 이용한 경로계획 방법과 비트맵을 이용한 충돌회피 방법을 제안하였다.

본 논문에서는 미지의 환경에서 미지의 장애물을 회피할 수 있는 스캔 코드법을 이용한 장애물 회피 방법을 제안한다. 원형의 장애물 인식 윈도우 안에 검출된 장애물에 대한 정보를 스캔 코드로 변환하여 장애물 회피를 위한 조향각 결정에 사용한다. 또한, 스캔 코드를 로봇의 충돌을 방지하기 위한 속도 변환 방법에 사용한다.

2. 본 론

2.1 스캔코드법을 이용한 경로계획

2.1.1 스캔코드 표현법

로봇의 중심에 놓인 원형 장애물 감지 윈도우는 반경이  $W_R$ 이다. 이 것의 원점을 직교하는 수직선과 수평선으로 영역을 나누었을 때, I,IV분면을  $Q_R$ 영역, II,III분면을  $Q_L$ 이라고 한다. 로봇의 장애물 감지 센서(초음파 센서)

가 2n개 장착되어 있을 때, 한 개의 센서가 인식할 수 있는 각도를 기준으로 장애물 윈도우를 2n 등분한다. 수직선을 기준으로  $Q_R$ 은 시계방향으로 비트 번호가 증가하고,  $Q_L$ 은 반시계 방향으로 비트 번호가 증가한다.

$$Q_R = b_1 b_2 b_3 \dots b_n \quad (1)$$

$$Q_L = b_1 b_2 b_3 \dots b_n \quad (2)$$

윈도우의 중심에 m개의 등간격 동심원을 구성하여 로봇과 장애물간의 거리를 나타낸다. 최외곽 원을  $r^1$ , 최내각 원을  $r^m$ 으로 나타내기로 한다.  $j$ 번째 원에 장애물이 인식되면 임의의  $i$ 번째 비트  $b_i$ 가 값은  $b_i = j$ 가 된다.

장애물은 스캔코드에서 0이 아닌 비트가 1개 이상인 것으로 정의된다. 장애물을 정의하는 요소를  $B_j$ 라고 할 때, 이것은 장애물의 유무를 표현하는 id와 장애물의 비트 크기를 표현하는 count로 정의된다. 즉,  $B_j.id = \{obs$  또는 free)와  $B_j.count = \{연속된 비트의 수\}$ 로 표현된다. 어떤 한 장애물은 다음과 같이 표현할 수 있다.

$$B_j.id = obs \quad (3)$$

$$B_j.count = n(b_s b_{s+1} b_{s+2} \dots b_{s+l-1}) \quad (4)$$

$$(\forall b_i \neq 0 \quad s, l \text{ 은 정수})$$

여기서,  $n(\cdot)$ 은 연속된 비트 수를 구하는 함수를 의미한다. 장애물이 없는 경우는 스캔코드에서 0인 비트의 수가 1개 이상일 때이다.  $B_j.id$ 와  $B_j.count$ 는 다음과 같이 정의된다.

$$B_j.id = free \quad (5)$$

$$B_j.count = n(b_s b_{s+1} b_{s+2} \dots b_{s+l-1}) \quad (6)$$

$$(\forall b_i = 0 \quad s, l \text{ 은 정수})$$

2.1.2 조향각의 계산

목표 지점을 향해 로봇이 주행하는 중에 윈도우 내에 장애물이 인식되지 않는 경우는 다음의 식(7)에 따라 로봇의 목표지점과 현재 위치간의 조향각이 결정된다.

$$\delta_i = \tan^{-1} \frac{(y_i - y_r)}{(x_i - x_r)} \quad (7)$$

여기서, 조향각  $\delta_i$ 는 직각 좌표계 상에서 로봇의 현재위치와 목표지점간의 각도이고,  $y_i$ 와  $x_i$ 는 목표지점의  $(x, y)$ 좌표 값이며,  $y_r, x_r$ 는 각각 로봇의 현재 위치 좌표값이다. 로봇의 중심점과 목표지점간의 거리는

$$d_i = \sqrt{(x_i - x_r)^2 + (y_i - y_r)^2} \quad (8)$$

이다.

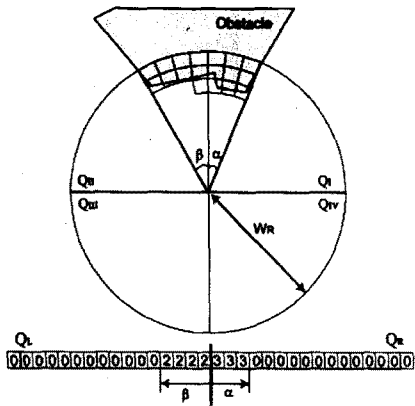


그림 1. 장애물 윈도우와 인식된 스캔코드

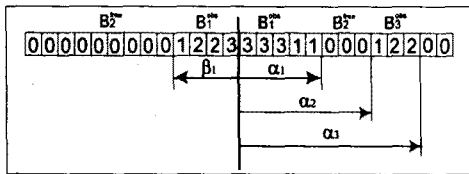


그림 2. 장애물이 인식된 스캔코드의 예

$B_j$ 가 차지하는 있는 공간에 대한 각도를  $\alpha_j$ 라고 표기할 때, 이것은 다음과 같이 계산할 수 있다.

$$\alpha_j = a_{unit} * B_j.count \quad (\text{degree}) \quad (9)$$

여기서,  $a_{unit}$ 는 한 개의 비트에 의한 단위각을 나타낸다.  $j \neq 1$ 인 경우는 현재값  $j$  이전의  $B_i.id = \text{free}$ 인 것의  $B_i.count$ 값이 로봇이 통과 가능하지 않다는 것을 의미하므로 이 경우에 각도는 다음과 같이 계산된다.

$$\alpha_j = a_{unit} * \sum_{i=1}^j B_i.count \quad (\text{degree}) \quad (10)$$

$Q_R$ 과  $Q_L$ 에서의  $B_j$ 를 각각 구분하기 위하여  $B_j^R$ 과  $B_j^L$ 과 같이 뱃침자를 붙이기로 한다.  $Q_R$ 과  $Q_L$ 의 전체 스캔코드에서 로봇이 통과 가능한 영역을 구하고 목표지점을 향해 이동하기 위한 조향각을 아래의 절차로 구하기로 한다.

절차 1.  $B_j^R.id = \text{obs}$ 이거나  $B_j^L.id = \text{obs}$

①  $j=1$  부터  $n$  까지  $B_j^R.id = \text{free}$ 중에서  $B_j^R.count \geq B_{safe}$ 인 것을 찾는다.

②  $j=1$ 부터  $n$ 까지  $B_j^L.id = \text{free}$ 중에서  $B_j^L.count \geq B_{safe}$ 인 것을 찾는다.

③ 식(9)에 의해  $B_j^R.id = \text{free}$ 에 의한  $\alpha_j$ 를 계산하고,  $B_j^L.id = \text{free}$ 에 의한  $\beta_j$ 를 각각 계산한다.  $\alpha_j$ 와  $\beta_j$ 를 다음식에 적용하여 조향각  $\delta$ 를 결정한다.

$$\delta = \min(\alpha_j, \beta_j) + \delta_{safe} \quad (11)$$

절차 2.  $B_j^R.id = \text{free}$ 이고  $B_j^L.id = \text{free}$ 인 경우

1)  $B_j^R.count + B_j^L.count \geq B_{safe}$ 을 만족할 때, 다음의 조건에 따라 조향각이 결정된다.

①  $\alpha_j \geq \frac{1}{2} \delta_{safe}$ 이고  $\beta_j \geq \frac{1}{2} \delta_{safe}$ 이면  
 $\delta = \delta_i$

②  $\alpha_j > \frac{1}{2} \delta_{safe}$ 이고  $\beta_j < \frac{1}{2} \delta_{safe}$ 이면  
 $\delta = \delta_i + (\delta_{safe} - \beta_j)$

③  $\alpha_j < \frac{1}{2} \delta_{safe}$ 이고  $\beta_j > \frac{1}{2} \delta_{safe}$ 이면  
 $\delta = \delta_i - (\delta_{safe} - \alpha_j)$

2)  $B_j^R.count + B_j^L.count < B_{safe}$  일 때, 절차 1의 방법을 따른다.

### 2.1.3 충돌방지를 위한 안전영역 계산

장애물 감지 윈도우 영역의 반경은  $W_R$ 이다. 주어진 시간동안 로봇이 최대속도로 직진 주행 시 (즉,  $w=0$ 일 때), 도달할 수 있는 거리는 반경  $\rho_{max}$ 로 표시되는 영역이다.

$$\rho_{max} = v \cdot t + R_p \quad (12)$$

여기서,  $R_p$ 는 로봇의 몸체의 반경을 나타낸다. 로봇이 회전 운동을 할 때 (즉,  $w \neq 0$ 일 때) 회전하고 난 후의 로봇은 임의의 반경  $\rho_w$ 안에 놓이게 된다.  $\rho_w$ 는  $w$ 값과 속도에 따라  $\rho_{max}$ 보다 작은 다른 값을 갖게 된다.

로봇이 최대속도  $v = v_{max}$ 로 최대 회전각속도  $w = w_{max}$ 로 회전할 때, 회전 반경  $R$ 는 다음과 같다.

$$R = \frac{v_{max} \cdot t}{w_{max}} \quad (13)$$

로봇이 직각좌표 점  $(x, y)$ 에 놓여있을 때, 각속도  $w$ 로 회전을 하고 난 후의 좌표  $(x', y')$ 에 위치하게 된다. 이 때, 반경  $\rho_w$ 는 다음과 같다.

$$\rho_w = \sqrt{x'^2 + y'^2} + R_p \quad (14)$$

속도 영역  $\rho_w$ 내에 장애물이 감지된 경우, 로봇의 회전 운동에 따른 충돌이 발생할 수 있다. 충돌을 방지하기 위해서는 속도를 줄여야 한다. 속도 영역  $\rho_w$ 에 장애물이 인식되지 않는 경우, 로봇은 최대속도  $v_{max}$ 로 주행한다. 속도 영역  $\rho_w$ 내에서 장애물이 윈도우의 원점에 가장 근접한 좌표  $(x_{obs}, y_{obs})$ 에 인식된 경우 속도 영역의 반경은 다음과 같이 구할 수 있다.

$$\rho_{obs} = \sqrt{x_{obs}^2 + y_{obs}^2} \quad (15)$$

속도 영역  $\rho_w$ 는  $\rho_w \leq \rho_{obs} \leq \rho_{max}$  범위에 있어야 한다. 그러므로, 속도  $v$ 는

$$v = v_{max} \left(1 - \frac{\rho_w}{\rho_{max}}\right) \quad (16)$$

가 된다. 로봇의 최저 속도를  $v_{min}$ 이라고 할 때, 로봇의 속도와 회전 반경은 항상 다음 범위의 값이 된다.

$$v_{min} \leq v \leq v_{max} \quad (17)$$

$W_R = m$ 일 때,  $\rho_{max}$ 는 일정한 거리를 나타내므로 이 값을  $r^{max}$ 라고 놓고,  $\rho_w$ 에 해당하는 거리는  $r^w$ 라고 표기한다.  $\rho_{obs}$ 에 해당하는 거리는  $r^{obs}$ 라고 하자.  $w = +a$ 일 때,  $Q_R$ 에

인식된 비트군  $B_j, id=obs$ 가 장애물이므로 이 영역에 대해서 비트 값을 조사하여 최고치 값을  $b_j^{max}$ 라고 하였을 때,

$$b_j^{max} = \max(b_1, b_2, b_3, \dots, b_n) \quad (18)$$

가 되고, 이 값으로  $\rho_{obs}$ 는 다음과 같은 값이 된다.

$$\rho_{obs} = b_j^{max} = r^{obs} \quad (19)$$

이 때, 안전한 회전을 위한 회전 반경  $\rho_w$ 는  $\rho_w \leq \rho_{obs} \leq \rho_{max}$  범위에 있어야 하므로 이것을 스캔코드의 표현으로 다음과 같이 나타낼 수 있다.

$$\rho_w = m - (b_j^{max} + 1) \quad (20)$$

그러므로, 식(20)의 속도식은 다음과 같이 나타낼 수 있다.

$$\begin{aligned} v &= v_{max} \left(1 - \frac{\rho_w}{\rho_{max}}\right) \\ &= v_{max} \left(1 - \frac{m - (b_j^{max} + 1)}{r_{max}}\right) \end{aligned} \quad (21)$$

## 2.2 시뮬레이션

로봇은 36개의 초음파 센서를 갖추고 있고, 이것은 모두 인식하는데 0.4초가 걸린다. 로봇의 최고속도는  $v_{max} = 0.95(m/sec)$ 이고 최저속도는  $v_{min} = 0.1(m/sec)$ 이다. 최대 각속도는  $\dot{\omega}_{max} = 1.2(rad/sec^2)$ 이다. 원형 윈도우의 인식반경은  $W_r = 2.5(m)$ 이고 각 비트당 각도는  $\alpha_{unit} = 10^\circ$ 이다.

그림 3은 사각형 장애물 7개와 원형 장애물 1개로 구성된 환경에서 start#1-#4까지의 시작점에서 목표점까지의 주행과정을 동시에 나타낸 것이다. 장애물과의 충돌회피를 위한 안전거리를 유지하며 최단거리로 주행하고 있다. 이 시뮬레이션에서 로봇의 초기 자세는 목표점에 대해서  $270^\circ$ 이다. 그림 4는 4개의 사각형과 1개의 원형 장애물로 구성된 밀집된 환경에서의 주행 시뮬레이션 결과이다. 같은 환경 하에서 각각 다른 3개의 시작점과 목표점으로 이동한 결과를 나타내었다. 로봇은 안전한 경로를 선택하여 장애물 회피를 하면서 목표지점에 정확하게 도달하였다. 이 시뮬레이션에서는 start #1-#3 모두 로봇의 초기자세는 목표지점에 대해서  $90^\circ$ 인 상황에서 주행하였다.

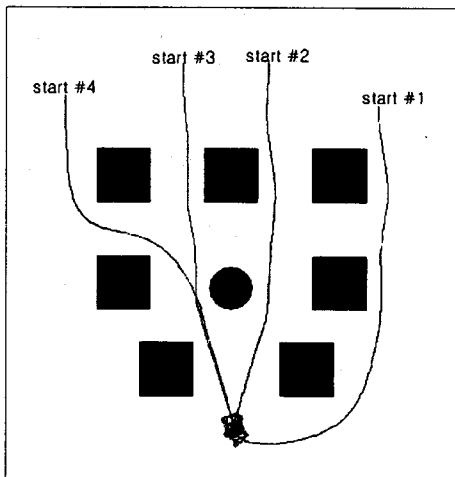


그림 3. 시뮬레이션 1

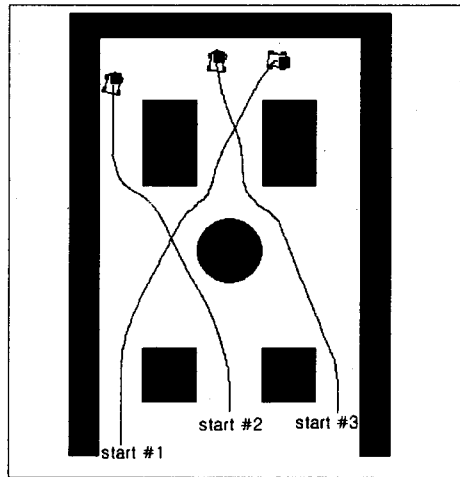


그림 4. 시뮬레이션 2

## 3. 결 론

부분적인 미지의 환경이나 완전한 미지의 환경에서 이동로봇을 목표지점까지 안전하게 이동시키는 것은 매우 중요한 문제다. 이를 실현하기 위하여 본 논문에서는 센서에서 인식된 장애물에 대한 정보를 스캔코드로 변환하여 장애물 회피를 위한 조향각 결정에 사용하였고, 장애물과 충돌을 방지하기 위한 로봇의 속도 변환에 사용하였다.

본 논문에서는 제안한 스캔코드를 이용한 장애물 회피 방법은 이동로봇에 일반적으로 많이 사용하는 초음파 센서에 적용하기 쉬우며, 간단한 계산으로 조향각을 결정할 수 있는 장점이 있다. 회전하는 로봇의 반경내에 장애물이 존재하는 경우는 조향각과 각속도를 조절하여 안전한 경로를 선택하여 충돌회피를 할 수 있었다. 미지의 환경에서 미지의 장애물을 회피하는 경우의 시뮬레이션을 수행하여 장애물 회피의 방법으로 유용하다는 것을 입증하였다.

## (참 고 문 헌)

- [1] Yuan F Zheng, Recent Trends in Mobile Robots, World Scientific, pp 75-117, 1993.
- [2] O. Khatib, "Real-time Obstacle Avoidance for Robot Manipulator and Mobile Robots", The International Journal of Robotics Research, Vol. 5, No. 1, pp 90-98, 1986.
- [3] Y. Koren and J. Borenstein, "Real-time Obstacle Avoidance for Fast Mobile Robots in Cluttered Environment", In Proc. IEEE Int. Conf. Robotics and Automation, Vol CH-2876, pp 572-577, 1990.
- [4] Y. Koren and J. Borenstein, "The Vector Field Histogram - Fast Obstacle Avoidance for Mobile Robots", IEEE Transaction on Robotics and Automation, Vol. 7, No. 3, pp 278-288, 1991.
- [5] Dieter Fox, Wolfram Burgard, and Sebastian Thrun, "The Dynamic Window Approach to Collision Avoidance", IEEE Robotics and Automation Magazine, Vol. 4, No. 1, pp 23-33, 1997.
- [6] Z. X. Li. and T. D. Bui, "Robot Path Planning using Fluid Model", Journal of Intelligent and Robotics Systems, Vol. 21, pp 29-50, 1998.