

윈도우 분석법을 이용한 이동로봇의 장애물 회피와 경로계획

조 규 상

동양대학교 컴퓨터공학부
gyusang@phenix.dyu.ac.kr

FAX : +82-572-630-4057 TEL : +82-572-630-1119

Obstacle Avoidance and Path Planning of Mobile Robot using Window Analyzing Method

GyuSang Cho

School of Computer Engineering, DongYang University
gyusang@phenix.dyu.ac.kr

FAX : +82-572-630-4057 TEL : +82-572-630-1119

요약

본 논문에서는 지역최소점에 도달하는 문제가 발생하지 않으며 빠르고 간편한 방식으로 로봇의 장애물 회피를 할 수 있는 윈도우 분석법을 이용한 새로운 장애물 회피와 경로계획 알고리즘 알고리즘을 제안한다. 미지의 경로상에서 장애물 윈도우에서의 특징을 이용하여 안전한 주행 경로를 선택하며, 장애물 회피를 한다. 시뮬레이션의 결과를 통하여 이 방법의 신뢰성과 우수성을 입증한다.

Abstract

In this paper, a new on-line local path planning algorithm without local minima problem is proposed. This method, called Window Analysis Method (WAM), is simple and fast, and it helps a robot take a safe path avoiding obstacles in unknown environment. WAM has simulated on the mobile robot to demonstrate its reliability and feasibility.

1. 서론

이동로봇에 있어서 목표지점을 향한 정확한 주행과 주행중에 발생하는 장애물에 대한 실시간 장애물 회피는 주요한 문제이다. 그 동안 발표된 로봇의 동작계획에 대한 많은 연구들은 미리 알고 있는 작업공간 안에서 고정 장애물에 대한 회피방법이 주류를 이루고 있다. 이러한 방법들은 미리 주어진 작업 공간 지도(workspace map)를 이용하여 최적한 이동로봇의 주행경로를 생성할 수 있었다. 그러나 실제적인 환경에서는 이동로봇의 경로상의 변화(장애물 나타남)가 생길 가능성이 있고, 주어진 공간지도상의 정보만으로는 로봇의 완전한 주행에 대한 계획을 할 수 없는 경우가 있다. 경로상에 놓여있는 예상치 못했던 장애물로 인해서 로봇이 멈추게 되는 경우가 생기는 것을 피해야 한다. 그러므로 미지의 환경에서 미지의 장

애물에 대한 회피를 할 수 있는 동작 계획방법이 필요하다.

WFM(Wall Following Method)[1]은 미리 정해진 간격으로 벽면을 따라 수행하는 방법이다. 만일 주행중 어떤 장애물을 만나면 또 다른 벽으로 간주하여 원래의 경로로 생각되는 경로를 찾을 때까지 우회한다. 이 방법은 기술적으로 용이하지만 다양성이 부족하고, 복도와 같은 특정한 환경에만 적합한 방법이다[2].

PFM(Potential Field Method)[3]은 장애물이 로봇에 척력을 발휘한다고 가정하고 목표위치가 인력을 발휘한다고 생각하는 방법이다. 이 방법은 로봇에 가까이 있는 장애물에 대해서 적용할 수 있고 매우 빠르게 이동경로에 대한 계획을 할 수 있다. Borenstein과 Koren[4]은 이 방법에서는 장애물간의 간격이 근접해 있는 경우에 지역최소점(local minima)에 의해서 트랩(trap)상태에 들어갈 경우가 있다고 밝혔다. 또한, 양쪽에 장애물(또는 벽면)이 있는 경우에 로봇의 동작 수행선이 진동을 하고, 장애물 근방에서 진동이 심하게 일어나며, 밀집된 장애물 사이로 주행하지 못하는 경우가 있다고 밝혔다. Koren과 Borenstein은 Virtual Force Field Histogram방법[5]을 개선한 Vector Field Histogram 방법[6]을 제안하였다. 이 방법에서는 연속적으로 갱신되고 생성되는 로봇의 환경 모델링을 하는데 Occupancy Grid 방법을 사용하여 연속적인 초음파 센서의 데이터를 갱신하여 불발성 잡음에 대처할 수 있도록 하였고, 점유된 격자의 정보를 이용하여 로봇이동의 방향과 속도를 계산하는데 사용한다. 이 방법에서의 단점은 길끝(dead-end)상황에서의 트랩상태에 들어갈 경우가 있고, 트랩상태에서 제자리서 순환하는 동작(cyclic behavior)을 나타내는 경우가 있다.

최근에 Fox등은[7] 이동로봇의 동작 다이내믹스(motion dynamics)가 직접 유도되기 때문에 로봇의 관성을 고려하여, 고속으로 장애물이 밀집된 환경에서도 안전하게 제어될 수 있는 Dynamic Window 접근법을 이용하여 이동로봇의 충돌회피방법을 제안하였다.

이동로봇의 경로를 계획하는데 있어서 주요한 점

은 고속으로 주행가능하며, 장애물회피나 주행중에 트랩상태에 들어가지 않아야 하며, 최적인 경로로 로봇의 주행이 이루어져야 한다.

본 논문에서는, 지역최소문제가 발생하여 트랩상태에 빠지지 않고 우수한 주행성능을 보이는 윈도우 분석법(Window Analyzing Method)을 이용한 장애물 회피와 경로 계획을 제안한다. 이 방법에서는 조향영역, 속도영역, 보조영역으로 구성된 윈도우내에 감지된 장애물의 점유된 형태에 따라 조향각과 주행속도를 결정하며, 또한 이 정보들을 이용하여 장애물 회피를 위한 주행의 방향을 결정한다. 장애물 회피와 경로계획에 대한 시뮬레이션의 결과를 통해서 이 방법의 효용성과 타당성을 입증하기로 한다.

2. 장애물에 대한 표현 방법

이동 로봇에 많이 사용되는 초음파 센서는 방향성에 둔감하고 반사현상이 있어 잘못된 측정을 하는 경우가 많이 있다는 단점이 있다. 이런 영향을 줄이기 위해 Elfes[8]는 Certainty Grid방법을 제안하였다. 그는 이 방법을 오프라인 전역 경로 계획에 사용하였으며, Moravec과 Elfes[9]는 map-building에 이 방법을 사용하였다. Borestein과 Koren[4,5,6]의 VFH 방법에서도 이것을 사용하여 장애물에 대한 표현을 하였다. 이와 유사한 개념으로 Yun[10]등은 확률을 이용한 장애물을 표현하였다.

Certainty Grid 방법은 2차원의 로봇의 작업 영역을 작은 셀로 구성된 영역으로 분할하며, 각 셀은 임의의 크기로 결정한다. 각 셀들은 C(i,j)로 나타낸다. 이 셀안에 장애물이 감지되면 C(i,j)에 기록한다. 방사형으로 주사하는 초음파 센서의 특징으로 인해서 센싱범위내의 장애물만 인식되고, 로봇트가 이동중에 계속적인 센싱을 하기 때문에 같은 위치에 해당하는 장애물이 여러번 감지된다. 감지된 장애물이 같은 위치에 있는 경우에 C(i,j)의 값을 증가시켜 셀의 가중치를 증가시킨다. 인식된 데이터들로 전체의 2차원 작업영역을 구성하게 된다. 같은 위치에 대한 여러번의 센싱으로 인해서 잡음이 아닌 실제 존재하는 장애물인 경우는 셀의 가중치가 높게 나타난다. C(i,j)의 값이 기준치 이상인 경우는 장애물이 있는 것으로 간주한다. 이 방법을 사용하면 초음파 센서의 부정확성으로 인해서 오인된 장애물에 대한 신뢰성 있는 인식을 할 수 있다.

3. 윈도우 분석법에 의한 장애물 회피와 경로 계획

본 논문에서 제안한 로봇의 장애물 회피와 경로 계획은 윈도우 분석법(Window Analyzing Method)이라 한다. 이 방법은 임의의 크기의 정방형 Certainty Grid 윈도우 안에 있는 장애물에 대한 정보를 이용한다. 윈도우내의 장애물 데이터들의 분포와 위치, 점유량에 따라 로봇의 속도와 방향이 결정된다.

3.1 윈도우의 구성

윈도우 내의 센싱된 정보는 셀 단위로 표현하며, 2장에서 설명한 Certainty Grid방법을 사용한다. 윈도우

의 크기는 NxN(N은 임의의 정수)으로 설정한다. 전체

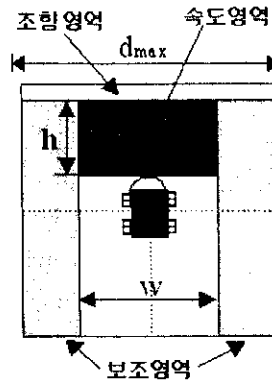


그림.1 윈도우 구성

윈도우는 로봇의 조향을 결정하는 조향영역, 장애물과의 관계에 의한 속도를 결정하기 위한 속도영역, 그리고 경로계획에 사용되는 보조영역으로 구성된다(그림 1). 윈도우의 크기(N)는 회전할 수 있는 최소의 범위 이상으로 설정하며, 센싱가능 범위내의 값이어야 한다. 이 범위를 너무 크게 정하면 전체적인 처리량이 늘어나기 때문에 효율이 낮아

지는 경향을 나타낸다. 윈도우의 중심축의 위치는 로봇의 중심축과 일치한다.

3.2 조향각의 결정

목표 지점을 향한 로봇의 주행 중에 윈도우내에 장애물이 감지되지 않는 경우는 식 (1)에 따라 로봇의 조향각이 결정된다.

$$\theta_r = \tan^{-1} \left(\frac{y_t - y_r}{x_t - x_r} \right) \quad (1)$$

여기서, θ_r 는 로봇과 목표지점과의 조향각이고, y_t 와 x_t 는 각각 목표지점의 (x,y)좌표값이고 y_r, x_r 는 각각 주행중인 로봇의 현지점의 (x,y)좌표계 값이다. 장애물이 윈도우내에 있는 경우는 다음 식에 따라 조향각이 결정된다.

$$\theta_r = \theta_{\max} \left(\frac{d}{d_{\max}} \right) \quad (2)$$

여기서 θ_{\max} 는 로봇트가 회전할 수 있는 최대각이다.(본 논문의 경우는 $\theta_{\max} = \pm(\pi/2)$ 이다) d는 윈도우 내에 감지된 장애물의 셀단위의 길이이고, d_{\max} 는 윈도우의 조향영역의 최대 길이이다.

3.3 속도의 결정

로봇의 이동속도는 속도영역에 점유된 장애물에 대한 계산을 다음 식에 따라 결정된다. 어떤 임계치 이상인 셀 $C(i,j) > Th$ 인 경우에 대해서 점유된 장애물 셀의 전체수를 W_s 할 때,

$$W_s = \sum_{i=0}^m \sum_{j=0}^n C(i,j) \quad (3)$$

가 된다. 속도는 다음을 식에 따라 계산된다.

$$v_r = c_v \cdot v_{\max} \left(1 - \frac{W_s}{W_{\max}} \right) + v_{\min} \quad (4)$$

여기서 c_v 는 속도계수이고, v_{\max} 와 v_{\min} 는 각각 로봇의 최대속도와 최저속도이다. W_{\max} 는 속도영역의 최대 크기이다.

3.4 경로의 결정(경로계획)

경우 I 최대각 회전(정면 장애물)

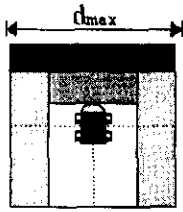


그림 2 최대각회전

윈도우의 조향영역 전면에 걸쳐 장애물이 나타나는 경우는 더 이상 로봇이 진행할 수 없는 경우이다. 그림 2의 경우는 d_{max} 에 의하여 최대회전하고, 방향은 목표지점이 놓인 곳이다.

$$\theta_r = \theta_{max} \quad (5)$$

경우 II 최대각 회전(막다른 골목)

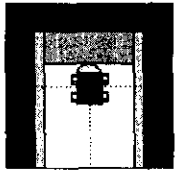


그림 3 막다른 골목

장애물이 그림 3과 같은 형태로 감지된 경우는 좁은 길의 막힌 끝에서 발생한다. 이 경우는 경우 I에서 적용되는 최대각 회전이 적용된다. 그러나 이 경우는 알려진 길이 나올 때까지 회전한다.

$$\theta_r = \theta_{max} \quad (6)$$

경우 III 모서리 회전

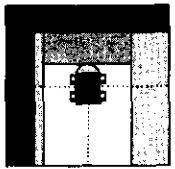


그림 4 모서리

그림 4의 경우는 전면과 한 측면이 막혀있는 경우이다. 이 경우는 열린진 측면으로 최대회전각으로 회전을 한다. 회전을 하고 난 후에는 식(1)-(4)가 적용이 되며, 주행 중에 다른 경우들이 적용된다.

경우 IV 좁은 길

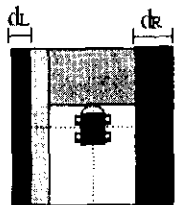


그림 5 좁은 길

그림 5의 경우는 좁은 길을 통과할 때 나타나는 윈도우 형태이다. 우측의 보조영역에서 측정된 d_R 값과 좌측의 보조영역에서 측정된 d_L 값의 차이에 따라 조향각이 결정된다. $d_R - d_L$ 값을 계산한 후에 이 값에 따라 조향각이 결정되며 계속적인 좁은 길이 계속적으로 이어지는 경우는 길의 중앙으로 로봇이 주행하게 된다.

경우 V 회전

양쪽에 벽면 장애물이 그림 6과 같은 형태로 놓여진

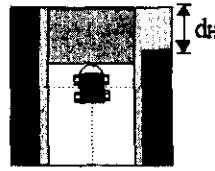


그림 6 회전

경우는 경우 IV과 같은 좁은 복도길을 주행하고 난 후에 나타날 수 있는 경우이다. 우측벽면이 윈도우 하방으로 높이 d_H 만큼 낮게 감지되었는데 이 경우는 목표지점의 위치에 따라 다른 조향각을 갖게 된다. 즉, 목표지점이 높이 h 로 인해서 회전하는 각보다 작은 경우는 목표지점의 각만큼 회전하

며, 목표지점이 회전각보다 큰 각의 범위에 놓여 있는 경우는 높이 d_H 에 의한 회전각 성분만큼 회전을 한다. 또한, 목표지점이 로봇의 좌측에 놓여진 경우라면 진행방향에 대한 변경없이 직선으로 정속 주행을 한다.

$$\theta_r = \theta_H \quad (\text{if } \theta_i > \theta_H)$$

$$\theta_r = \theta_i \quad (\text{if } \theta_i < \theta_H) \quad (7)$$

경우 VI 통과할 수 없는 좁은 길

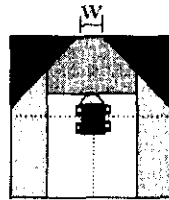


그림 7 좁은길(2)

그림의 상황은 열린진 길은 있지만 로봇이 진행하기에는 너무 좁은 길인 경우이다. 이 길로 진입을 하면 벽면과 이 경우는 경우 I에 준하는 회전을 한다.

$$\theta_r = \theta_{max} \quad (8)$$

4. 시뮬레이션

시뮬레이션 환경은 6m x 4m의 공간에서 로봇의 주행속도는 최대 초속 50cm/sec이다. 로봇에는 360도 방향으로 초음파 센서 데이터를 입력받는다. 전체 윈도우의 크기는 50x50으로 설정하였다. 로봇의 최대 회전각은 $\pm 45^\circ$ 이다.

그림 8의 시뮬레이션 1에서 시작지점은 start이고 목표지점은 end이다. #1지점에서는 경우 IV가 적용이 되어 길의 중앙으로 통과하고 있다. #2지점에서는 경우 V가 적용이 되어 목표지점을 향해 우회전을 한다.

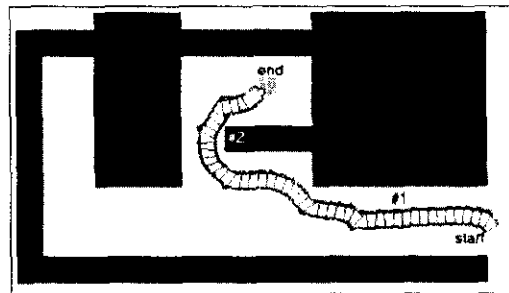


그림 8 시뮬레이션 1

그림 9의 시뮬레이션 2에서 #3지점은 윈도우의 조향 영역 전면에 장애물이 인식이 되고 로보트가 통과하기 충분한 간격이므로 이곳을 **경우 III**을 적용하여 통과하게 된다. #4와 #5에서는 지점에서는 **경우 V**가 적용

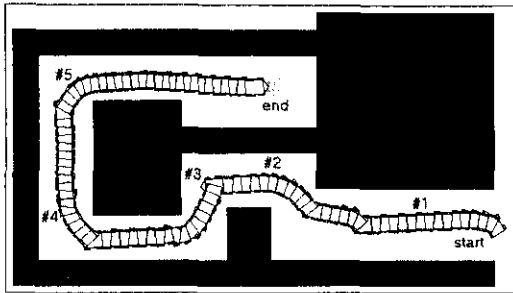


그림 9 시뮬레이션 2

되는 지점이다.

그림 10의 시뮬레이션3에서, #1의 좁은 길에서는 **경우 IV**가 적용이 된다. #3에서는 **경우 III**이 적용이 된다. #2지점에서 PFM을 이용한 주행에서는 트랩현상을 나타내지만, 본 논문의 WAM을 적용한 경우에는는 목표를 향해 정확한 주행을 한다.

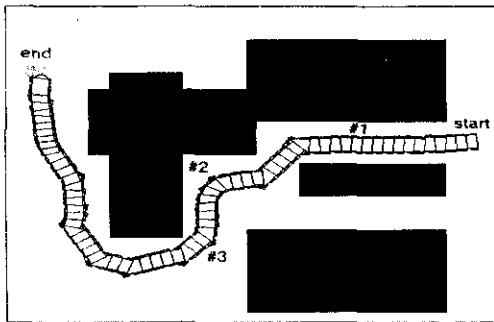


그림 10 시뮬레이션 3

그림 11의 시뮬레이션4는 그림 10의 시뮬레이션 3과 같은 환경이지만 출발지점을 달리하여, 다른 경로로 목표지점에 도달한다. #1 지점의 좁은 길을 통과하여 #2지점에 도달하게 된다(조향영역 전면에 장애물이 인식되지 않으므로). 이 지점에서는 **경우 IV**가 적용이 되어 그림과 같은 형태로 회전을 하여 #3지점을 통과하여 목표지점에 도달한다. PFM을 적용한 경우에는서는 이 지점에서 트랩상황에 빠지게 되지만, WAM을 적용한 경우는 트랩이 발생하지 않는다.

5. 결 론

본 논문에서는 윈도우 분석법(WAM)을 이용한 이동로봇의 장애물 회피 방법과 경로 계획방법에 관하여 논하였다. 이 방법에서는 주행중에 센싱 데이터값을 Certainty Grid Map으로 구성하였고, NxN 크기의 윈도우 영역을 설정하여 윈도우에 구성된 데이터들로부터 주행방향과 속도에 관한 정보를 얻어서 장애물 회피와 경로계획을 하였다. 시뮬레이션의 결과를 통해서

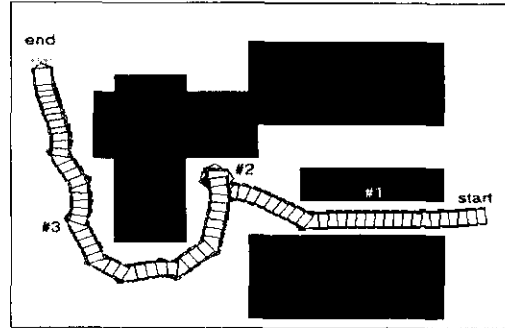


그림 11 시뮬레이션 4

이 방법이 이동로봇의 주행에 적합한 방법이라는 것을 알았으며, 다른 방법들에 비하여 트랩상황에 빠져 들어가는 확률이 적었으며, 또한 윈도우 내의 셀에 대한 간단한 계산을 하기 때문에 빠른 시간내에 주행방향과 속도를 결정할 수 있었다.

경험적으로 선택한 윈도우의 크기와 윈도우 내에 감지된 장애물의 위치, 높이 등의 정보를 통한 주행방향과 속도의 결정방법은 주행 성능에 영향을 주기 때문에 이것에 대한 정량적인 분석이 필요하다. 또한, 모든 장애물은 정적인 것으로 국한하였으며, 크기가 작은 장애물에 대한 경우에 대하여 고려하지 않았기 때문에 이에 대한 추가적인 연구가 수행되어야 한다.

6. 참고문헌

- [1] Yuan F Zheng, Recent Trends in Mobile Robots, World Scientific, pp 75-117, 1993.
- [2] H.P. Moravec, and A. Elfes, "High Resolution Maps from Wide Angle Sonar", IEEE Conf. on Robotics and Automation, Washington D.C., pp. 116-121, 1985.
- [3] 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.
- [4] Y. Koren and J. Borenstein, "Potential Field Methods and their inherent limitations for mobile robot navigation", In Proc. IEEE Int. Conf. Robotics and Automation, April 1991.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] A. Elfes, "A Sonar-Based Mapping and Navigation System", Carnegie-Mellon University, The Robotics Institute, Technical Report, pp 25-30, 1985.
- [9] Davis Lee and Michael Recce, "Quantitative Evaluation of the Exploration Strategies of a Mobile Robot", The International Journal of Robotics Research, Vol. 16, No. 4, pp 413-447, Aug. 1997.
- [10] W.S. Yun, et al., "Dynamic Path Planning for Robot Navigation Using Sonar Mapping and Neural Networks", Journal of Dynamic System, Measurement, and Control, Vol. 119, pp 19-26, Mar. 1997.