

VFH(VectorFieldHistogram)을 이용한 4족 로봇의 장애물 회피

Obstacle avoidance using VFH(VectorFieldHistogram) in four-legged robot

정현룡*, 김영배**

(Jung, HyunRyong and Young-Bae Kim)

* 전남대학교 기계공학과(전화:(062)530-0209, 팩스:(062)269-4608, E-mail : jhr1021@hanmail.net)

** 전남대학교 기계공학과(전화:(062)530-1677, 팩스:(062)530-1689, E-mail : ybkim@chonnam.ac.kr)

Abstract : The vector field histogram(VFH) uses a two-dimensional Cartesian histogram grid as a world model. The VFH method subsequently employs a two-stage data-reduction process in order to compute the desired control commands for the vehicle. In the first stage the histogram grid is reduced to a one dimensional polar histogram that is constructed around the robot's momentary location. Each sector in the polar histogram contains a value representing the polar obstacle density in that direction. In the second stage, the algorithm selects the most suitable sector from among all polar histogram sectors with a low polar obstacle density, and the steering of the robot is aligned with that direction. We applied this algorithm to our four-legged robot.

Keywords : VFH, CV, OpenGL, Map-building, Path finding

I. 서 론

장애물 회피(Obstacle Avoidance)는 mobile robot 시스템의 성공 여부에 있어 매우 중요한 부분이다. 모든 mobile robot은 장애물을 인식하고 정지하는 간단한 알고리즘부터, 아주 정교한 알고리즘을 탑재하여 장애물을 회피하는 것까지 아주 다양하다. 후자의 경우는 장애물의 텁색뿐만 아니라, 여러 가지 계산으로 인해서 전자보다 더욱 복잡하게 된다. VFH(Vector Field Histogram)은 후자의 경우로 기존의 Potential Field Method, Virtual Force Field등의 방법보다 계산 양이 많이 줄어들어, 로봇이 실시간으로 장애물을 회피할 수 있다. 본 논문에서는 이러한 VFH[5]을 4족 로봇에 적용하여 테스트 하였다.

II. 이 론

1. HIMM(Histogrammic In-Motion Mapping)

HIMM은 장애물을 표시하는데 있어서 이차원 Cartesian histogram grid를 사용한다. 그리고 histogram grid의 각 cell들은 certainty value(CV)값을 갖는다.

Histogram grid는 개선되는 방법에 있어서 CMU의 certainty grid method[2][3][4]와는 다르다. CMU의 방법은 초음파 센서 영역내의 cell들 모두 개선시키는 방법을택하고 있다. 하지만 이러한 방법은 막대한 양의 계산 양을 요구하게 되고 실시간으로 map-building하는 것을 어렵게 만들게 한다. 반면에 HIMM은 센서에 의해서 검출된 거리 d 에 위치하는 cell의 값만을 증가시키게 된다. 계산 양이 줄어들게 됨으로 로봇이 움직

이는 동안 실시간으로 계속해서 map-building이 가능하게 된다. 그리고 센서에 의해서 선택된 cell의 CV값이 증가함과 동시에 주변의 cell들의 CV값도 동시에 증가하게 된다. 하지만, 주지해야 하는 점은 HIMM은 로봇의 정지해 있는 경우에는 로봇이 움직이는 경우보다 정확도가 떨어진다는 점이다.

센서에 의해서 선택된 cell의 증가치(I^+)는 3이며, CV의 최대 값(CV_{max})은 15이다. 그리고 선택된 cell과 센서사이에 놓인 모든 cell들은 $-1(I^-)$ 만큼 감소하게 된다(Fig. 1). CV의 최소값(CV_{min})은 0이다. I^+ 와 I^- 의 값은 각각 독립적으로 결정되게 된다. I^+ 의 값은 실험으로 결정하게 되며, I^- 의 값은 I^+ 보다 적은 값으로 결정하여야 한다. 한번의 센서 reading으로 하나의 cell을 증가하게 되지만, 다수의 cell들은 감소하게 되기 때문이다.

HIMM에서는 로봇이 움직이는 동안에 정확한 map-building을 하기 위해서 Growth Rate Operator(GRO)을 사용한다. GRO는 센서에 의해서 검출된 cell 주변에 CV값이 0보다 큰 cell들이 있을 경우는 현재 선택된 cell에 영향을 주도록 한다. 즉 선택된 cell의 주변 cell들에 CV값이 높으면, 현재 선택된 cell의 CV값도 높아지게 되는 것이다.

GRO를 사용함으로 로봇이 움직이는 동안 발생할 수 있는 익기 오류로 인한 잘못된 map-building을 피할 수 있게 된다. 또한 로봇이 빠르게 움직이는 동안 적은 수의 sampling 만으로도 map-building이 가능하게 된다. CV값의 계산 알고리즘을 표현하여 보면 아래 식

과 같다.

$$CV'_{i,j} = CV_{i,j} + I^+ + \sum_{p,q=-1}^{p,q=1} (w_{p,q} CV_{i+p, j+q}) \quad (1)$$

여기서,

$CV_{i,j}$: previous certainty value of cell
(i, j)

$CV'_{i,j}$: updated certainty value of cell
(i, j)

I^+ : constant increment ($I^+ = 3$)

w : weighting factor

Object	0 0 3 0	0.5 0.5 0.5	0 0 15 0
0 0 3 0	0.5 0.5 0.5	0.5 1 0.5	12 15 15 0
3 3 3 0	0.5 0.5 0.5	0.5 0.5 0.5	6 12 0 0
6 6 0 0	GRO Mask	0 0 0 0	0 0 0 0
0 0 0 0	a	b	c

그림 1. (a). 로봇이 움직이는 동안 하나의 장애물에 대한 CV값

(b). 3×3 mask for growth rate operator(GRO)

(c). GRO를 적용한 상태

Table 1은 그림1를 기초로한 CV값 계산 예이다.

Read		CV
1	$a = a + I^+$	=0+3
2	$a = a + I^+$	=3+3
3	$b = b + I^+ + \frac{1}{4}a$	=0+3+3
4	$b = b + I^+ + \frac{1}{4}a$	=6+3+3
5	$c = c + I^+ + \frac{1}{4}a + b$	=0+3+9
6	$d = d + I^+ + \frac{1}{4}a + b + c$	=0+3+15
7	$e = e + I^+ + \frac{1}{4}b + d$	=0+3+13.5
8	$f = f + I^+ + \frac{1}{4}e + f$	=0+3+15

Table 1 그림1의 CV 계산 예($CV_{max} = 15$, ()의 값은 임시 값)

2. VFH(Vector Field Histogram)

VFH는 Virtual Force Field(VFF)의 단점을 보완하기 위해서 만들어졌다.

VFH는 2단계로 이루어진 자료 간략화(two data reduction)가 있다.

첫 번째 단계는 histogram grid(C)의 active region(C*)을 polar histogram(H)로 다음과 같이 바꿔 준다. 먼저, histogram grid의 모든 active cell의 값을 obstacle vector로 사용하고, 방향(β)은 현재 로봇의 중심좌표(Vehicle Center Point)를 기준으로 결정된다.

$$\beta_{i,j} = \tan^{-1} \left(\frac{y_j - y_0}{x_i - x_0} \right) \quad (2)$$

$$m_{i,j} = (c_{i,j}^*)^2 (a - bd_{i,j}) \quad (3)$$

여기서,

a, b : Positive constants

$c_{i,j}^*$: Certainty value of active cell(i,j)

$d_{i,j}$: Distance between active cell(i,j) and the VCP

$m_{i,j}$: Magnitude of the obstacle vector at cell(i,j)

x_0, y_0 : Coordinates of active cell(i,j)

$\beta_{i,j}$: Direction from active cell(i,j) to the VCP

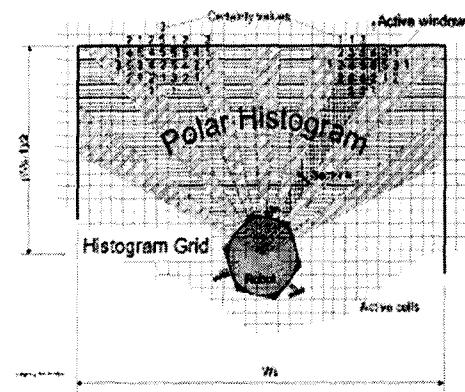


그림 2. Mapping of active cells onto the polar histogram

$c_{i,j}^*$ 은 사각영역이며, $m_{i,j}$ 는 $-d$ 에 비례한다. 그래서 장애물이 로봇에 가까울수록 $m_{i,j}$ 는 큰 값을 갖게 된다. 그리고 가장 멀리 떨어진 active cell의 $m_{i,j}$ 는 0이 된다.

d_{max} 는 active cell과 로봇의 중심(VCP)에서 가장 거리를 나타내며, $a - bd_{max} = 0$ 이다. 로봇의 중심을 기준으로 5° 간격으로 전 방향을 나누고 각 섹터는 $k = 0, 1, 2, \dots, n-1$ ($n=72$)라고 하자. 그리고 각 섹터에 대한 polar obstacle density는 h_k 라하고, 다음과 같이 계산 한다.

$$h_k = \sum_{i,j} m_{i,j} \quad (4)$$

그리고 이렇게 각 섹터 별로 계산된 POD(Polar Obstacle Density)에 임계값(threshold)을 적용하여 안전영역(safe area)와 비 안전 영역(unsafe area)으로 나눈다(그림 3).

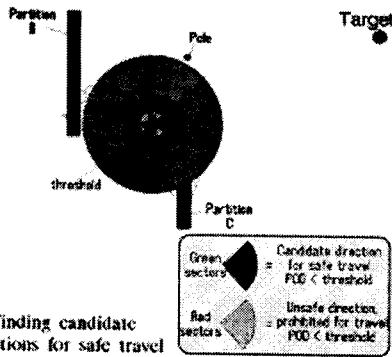


그림 3. A threshold on the polar histogram determines the candidate directions for subsequent travel.

이렇게 두 가지 영역으로 나누게 되면, 두 개 이상의 안전영역이 나오게 되는 경우가 발생하게 되는데, 이런 경우는 목표점에 가장 가까운 방향의 영역을 선택하도록 한다. 그리고 선택된 영역이 로봇이 통과할 수 있을 만큼의 공간인지를 판단하고, k_f (far border)와 k_n (near border)를 정한다.

이렇게 k_f 와 k_n 이 정해지면 로봇이 진행해야 할 방향 θ_{steer} 를 계산한다(그림 4).

$$\theta_{steer} = \frac{(k_n + k_f)}{2} \quad (5)$$

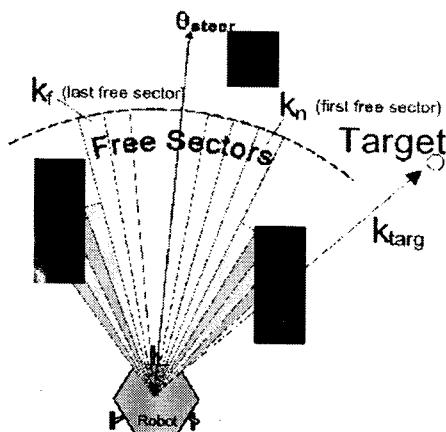


그림 4. Finding the steering reference direction θ when k_{targ} is obstructed by an obstacle

III. 연구 내용

실험은 자체 제작된 4족 로봇을 이용하여 실시하였으며, 로봇의 사양은 다음과 같다.

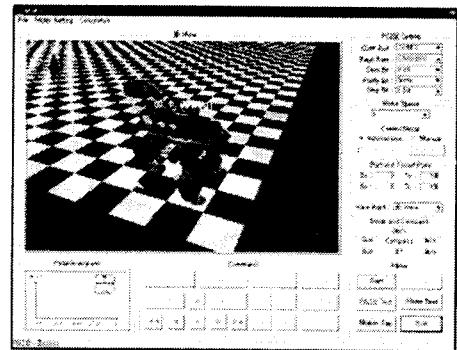


그림 5. 로봇 제어용 프로그램

모터제어를 위해서 PIC16F876이 사용되었으며, 5개의 초음파 센서를 사용하여 거리 측정을 하였다. 그리고 로봇으로부터 들어는 거리 정보를 바탕으로 맵을 그리기 위해서 로봇과 컴퓨터는 블루투스를 이용하여 통신을 하였다.

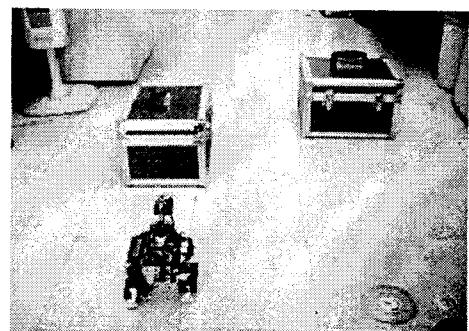


그림 6. 장애물 배치 상태

컴퓨터에서는 OpenGL[6]을 이용하여 맵을 표시하도록 하였다.(그림 5) 실험은 그림 6과 같이 장애물을 위치시키고 로봇이 이동하도록 하였다.

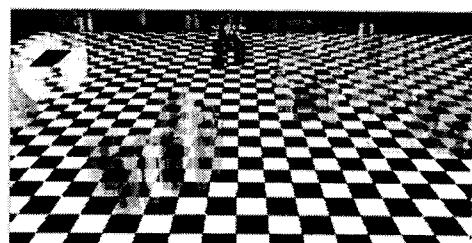


그림 7. 장애물 회피 및 map-building 결과

실험 결과는 그림 7과 같다. 그림에서도 알 수 있듯이 로봇이 장애물을 잘 회피하고 목적지까지 도착한 것을 알

수 있다. 그림 7에서 색깔을 가진 사각의 박스는 그 위치에 존재할 장애물의 CV에 따른 것이다.(Table 2)

Color	CV
Yellow green	0 ~ 2
Blue	3 ~ 5
Yellow	6 ~ 8
Pink	9 ~ 11
Red	12 ~ 15

Table 2 Cell colors according to the CV value

V. 결 론

VFH는 목표점까지 최적의 경로[7]를 찾는 방법이 아니라 빠른 속도로 이동하는 로봇을 위한 적합한 방법이다. 또한 장애물을 통과함에 있어서 멈추지 않는 장점이 있다. 또한 기존의 이론들[1][8]에서 해결하지 못한 출입구 형태나, 복도에서도 잘 이동할 수 한다. 본 논문은 기존의 mobile robot에 적용되었던 VFH를 관절 로봇에 적용하고 테스트하였다. 결과는 실험에도 보여주듯이 4족 로봇에서도 잘 동작함을 알 수 있었다. 하지만, map-building에 있어서는 mobile robot을 이용한 결과보다는 못하였다. 그러한 이유는 센서 값이 갱신되는 방법에 있다. Mobile robot의 경우는 움직이는 동안 계속해서 맵을 갱신하게 되지만, 관절로봇의 경우는 하나의 동작이 끝난 후에 거리데이터를 읽게 되어 맵의 정확도가 떨어지게 된다. 또한 localization이 해결되어야만 보다 정확한 map-building과 path finding이 이루어 질 것이다.

참고문헌

- [1]. Borenstein, J. and Koren, Y., "Real-time Obstacle Avoidance for Fast Mobile Robots." IEEE Transactions on Systems, Man, and Cybernetics, Vol. 19, No. 5, Sept/Oct 1989, pp. 1179-1187.
- [2]. Elfes, A, "Using Occupancy Grids for Mobile Robot Perception and Navigation." Computer Magazine, June, pp. 46-57 1989
- [3]. Moravec, H.P. and Elfes, A., "High Resolution Maps from Wide Angle Sonar." IEEE Conference on Robotics and Automation, Washington D.C., 1985, pp.116-121.
- [4]. Moravec, H.P., "Sensor Fusion in Certainty Grids for Mobile Robots." AI Magazine, Summer 1988, pp.61-74.
- [5]. Borenstein, J. and Koren, Y., 1991, "The Vector Field Histogram -- Fast Obstacle-Avoidance for Mobile Robots." IEEE Journal of Robotics and Automation, Vol. 7, No. 3, June 1991, pp. 278-288
- [6]. Andre LaMothe, "OpenGL Game Programming", pp.626-675, 2001
- [7]. Borenstein, J. and Koren, Y., "Real-time Obstacle Avoidance for Fast Mobile Robots." IEEE Transactions on Systems, Man, and Cybernetics, Vol. 19, No. 5, Sept/Oct 1989, pp.1179-1187
- [8]. Khatib, O., "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots." 1985 IEEE International Conference on Robotics and Automation St. Louis, Missouri, March 25-28, 1986, pp. 500-505.