

# Land Gambling DT를 이용한 원격 이동 로봇의 경로 계획

## Path Planning Using Land Gambling DTs for a Remote Mobile Robot

김상문, 이기성, 최경삼

홍익대학교 공과대학 전기제어 공학과

Sangmoon Kim, Keeseong Lee, Gyung-sam Choi

### Abstract

본 논문은 원격 제어가 CCD 카메라를 이용해 이동로봇의 상태를 파악해서 무선으로 이동로봇을 제어하고, 사용자는 인터넷을 통해 원격 제어기와 통신하면서 이동로봇을 조종하여 이동로봇이 장애물을 피해 가면서 원하는 위치까지 이동할 수 있게 하는 알고리즘을 제안한다. 제안하는 알고리즘인 Land Gambling DT를 통해 일단 경로계획을 하고 생성된 경로를 따라 이동로봇이 이동 중 장애물이 있는 경우 VFH를 이용하여 장애물을 피해 가는 것이다.

## I. 서론

본 논문에서 설정한 환경은 다음과 같다. 원격 제어기는 CCD카메라를 이용해 이동로봇의 상태를 파악하고, 무선으로 이동로봇을 원격 제어한다. 사용자는 인터넷을 통해 원격 제어기와 통신하면서 이동로봇을 조종한다. 본 논문은 이런 환경에서 이동로봇이 장애물을 피해 가면서 원하는 위치까지 이동할 수 있게 하는 알고리즘을 제안한다. 그 알고리즘은 다음과 같다. 거리변환(Distance Transform)[1,2]를 통해 일단 경로계획을 한 후 생성된 경로를 단순화시킨다(Land Gambling). 여기서 생성된 경로대로 이동 중 VFH(Vector Field Histogram)[3]을 이용해 장애물을 회피하게 된다. 이 알고리즘을

LGDT-VFH(Land Gambling Distance Transform - Vector Field Histogram)이라 부른다.

## II. 본론

### 1. DT(Distance Transform)

DT는 Binary Image Processing에서 형태 분석을 위한 도구로서 처음 사용되었다[1]. DT는 격자로 나누어진 공간에서 목표 위치부터의 거리를 전파 시켜서 공간 내의 어느 위치에서라도 목표점까지의 거리를 쉽게 알 수 있게 하는 방법이다. 이동로봇은 로봇 근처에 있는 여덟 개의 격자 중 목표에서 가장 가까운 격자로 이동하는 동작을 반복하면 결

국 목표점에 도달하게 된다. 그림1은 거리 변환의 예를 표시하고 있다. 각각의 Cell에 표시되어 있는 숫자는 그 지점에서 장애물까지 가는데 필요한 거리를 뜻한다. 여기서 거리는 수직·수평 이동은 3, 대각선 이동은 4이다. DT는 대단히 빠르고 경로가 존재하면 반드시 찾아내는 알고리즘이지만, 생성된 경로가 대각선이나 수직·수평 방향뿐이고 출발 장애물에 대처하지 못한다는 단점을 가진다.

33	30	27	24	21	18	15	12	9	6	3	3	6	
34	31	28	25	22	19	16	13	10	7	4	3	4	7
35	32								7	6	7	8	
36	35								10	9	10	11	
39	38								13	12	13	14	
42	41								16	15	16	17	
45	44	41	38	35	32	29	26	23	20	19	18	19	20
48	45	42	39	36	33	30	27	24	23	22	21	22	23
49	46	40	37	34	31	28	27	26	25	24	25	26	

그림 1. Distance Transform(DT)

## 2. Primitive DT

DT의 결과는 그림1에서 보인 것처럼 각각의 Cell에서 목표점에 이르는 거리이다. 그러나 이 거리는 실제 거리가 아니라 대각선이나 수직·수평 방향만을 사용했을 때의 거리이다. Cell에 실제의 거리를 채워 넣는 Primitive DT 알고리즘을 제안한다. 그 과정은 다음과 같다.

11.0	10.0	9.0	8.0	7.0	6.0	5.0	4.0	3.0	2.0	1.0	1.0	1.0	2.0
11.0	10.0	9.0	8.0	7.0	6.0	5.0	4.1	3.1	2.2	1.4	1.0	1.4	2.2
99.0	99.0									2.2	2.0	2.2	2.8
99.0	99.0									3.1	3.0	3.1	3.6
99.0	99.0									4.1	4.0	4.1	4.4
99.0	99.0									5.0	5.0	5.0	5.3
99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	6.0	6.0	6.0	6.3
99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	7.0	7.0	7.0	7.2
99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	8.2	8.0	8.0	8.2

그림 2. 첫 번째 과정을 거친 Cell

첫째 그림2와 같이 Cell들 중에 장애물에 충돌하지 않고 직선으로 목표점에 도달할 수 있는 Cell들을 찾아 목표점과 그 Cell 사이의

거리를 Cell에 채운다. 채워진 Cell들을 1번 Cell이라 한다. 둘째 첫째 과정에서 채워지지 않은 Cell중에서 장애물에 충돌하지 않고 직선으로 1번 Cell에 도달할 수 있는 Cell을 1번 Cell의 값과 그 Cell까지의 거리의 합의 최소 값으로 채운다. 1번 Cell을 0번 Cell이라 하고 채워진 Cell을 1번 Cell이라 한다. 셋째 Cell이 모두 채워질 때까지 둘째 과정을 반복한다. 이동로봇은 장애물과 충돌하지 않고 갈 수 있는 Cell 중에서 가장 숫자가 적은 Cell을 찾아서 계속 이동한다. 그림 3은 Primitive DT의 결과이다. Primitive DT는 가상 로봇의 Turn수도 작고 길어도 짧은 경로를 발생시키지만 알고리즘을 수행하는데 시간이 오래 걸린다는 단점을 가진다.

11.0	10.0	9.0	8.0	7.0	6.0	5.0	4.0	3.0	2.0	1.0	1.0	1.0	2.0
11.0	10.0	9.0	8.0	7.0	6.0	5.0	4.1	3.1	2.2	1.4	1.0	1.4	2.2
11.2	10.4									2.2	2.0	2.2	2.8
11.8	12.0									3.1	3.0	3.1	3.6
13.2	13.0									4.1	4.0	4.1	4.4
14.1	14.0									5.0	5.0	5.0	5.3
15.1	15.0	14.0	13.0	12.0	11.0	10.0	9.0	8.0	6.5	6.0	6.0	6.0	6.3
16.1	15.1	14.1	13.1	12.1	11.1	10.2	9.2	7.9	7.2	7.0	7.0	7.0	7.2
16.2	15.3	13.3	12.4	11.4	10.5	9.3	8.7	8.2	8.0	8.0	8.0	8.0	8.2

그림 3. Primitive DT의 결과

## 3. Fast Primitive DT

Primitive DT는 수행 시간이 오래 걸리는 알고리즘이다. Primitive DT의 속도를 개선하기 위한 Fast Primitive DT 알고리즘을 제안한다. Fast Primitive DT는 DT를 먼저 수행하고 DT에서 생성된 경로 주변에 대해서만 Fast Primitive DT를 수행하는 알고리즘이다. 이 알고리즘은 DT에서 생성된 경로와 Primitive DT에서 생성된 경로가 비슷하다는 것을 전제로 하므로 경우에 따라서는 Primitive DT와 다른 경로가 생성될 수도 있다. 그림4는 Fast Primitive DT의 결과이다.

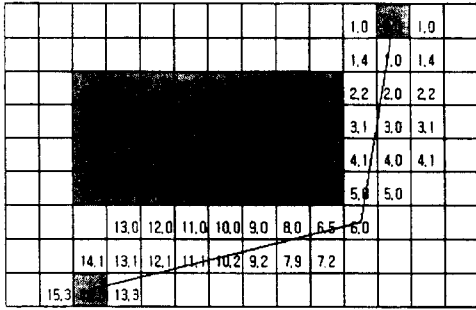


그림 4. Fast Primitive DT의 결과

#### 4. Land Gambling DT

Fast Primitive DT는 DT로 생성된 경로상의 값과 그 근처의 값을 고려한다. 따라서 Fast Primitive DT에서 발생된 경로는 모두 DT에서 발생된 경로의 근처가 된다. 그러면 DT로 발생된 경로만을 고려해도 비슷한 경로가 발생할 것이다. Primitive DT의 속도보다 더 개선하기 위해 DT로 생성된 경로에 대해서만 Primitive DT를 행하는 Land Gambling DT를 제안한다. 그림5는 Land Gambling DT의 결과이다.

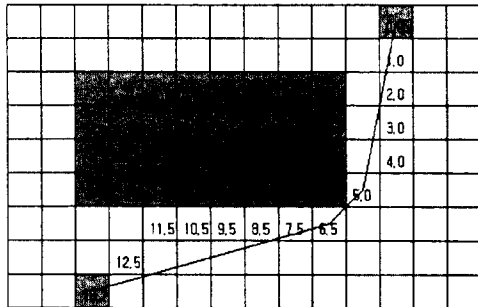


그림 5. Land Gambling DT의 결과

#### 5. VFH(Vector Field Histogram)

VFH[3]는 초음파 센서로 계측된 데이터를 바탕으로 CGM(Certainty Grid Map)을 만들고, CGM을 이용해서 그림6과 같은 Polar Histogram을 작성하고[3], Polar Histogram에서 일정한 Threshold를 정해 그 Threshold 이하의 값이 나오는 방향 중에서 목표점에 가장 가까운 방향으로 이동하는 알고리즘이다. VFH는 장애물을 부드러운 경

로로 비켜가고 돌발 장애물에도 강하지만 길을 찾지 못할 수도 있다는 단점을 가진다.

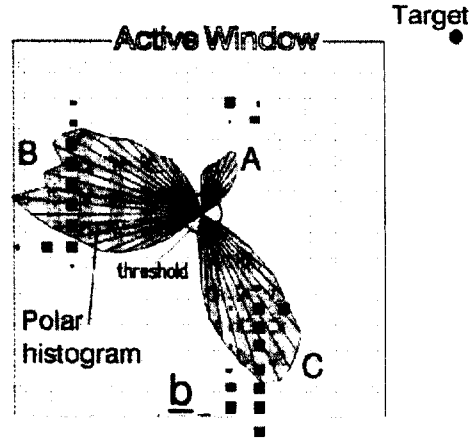


그림 6. Polar Histogram

그림 7은 Land Gambling DT에서 얻어진 경로를 VFH를 이용해 추적한 결과이다.

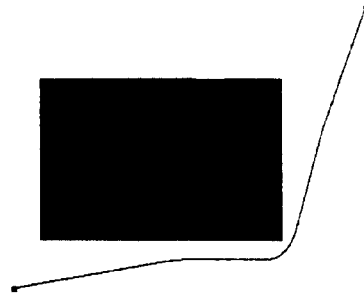


그림 7. Land Gambling DT의 VFH추적

#### 6. 실험결과

DT, Primitive DT, Fast Primitive DT, Land Gambling DT에 대해 비교 실험을 수행한 결과 표1과 표2와 같은 결과가 얻어졌다. 비교 대상은 시간, 거리, Turn횟수이다. 실험은 Java를 이용해 시뮬레이션 프로그램을 작성해서 14×9, 48×64 크기의 Map에서 수행했다. Primitive DT의 경우 64×48 Map에서 2시간 정도 걸렸으나, Fast Primitive DT의 경우 같은 경로가 생성되었고, 21초밖에 걸리지 않았다. Land Gambling DT의 경우 7초만에 경로를 발생 시켰다. 본 논문

서는 기존의 DT 알고리즘을 개선했고 개선된 DT 알고리즘이 빨리 이동로봇의 경로를 찾을 수 있음을 보여준다.

	시간(sec)	Length	Turn
DT	0.00	14.7	3
Primitive DT	1.70	14.3	1
Fast Primitive DT	0.33	14.3	1
SGA	5.83	15.12	4
Land Gambling DT	0.22	13.8	2

표 1. 14×9 Map에서 실험 결과

다음은 새로운 환경에서 실험한 결과이다. 그림 8, 그림 9, 그림 10, 그림 11, 그림 12는 각각의 알고리즘에 의해 생성된 경로이다. 표 2는 실험 결과이다.

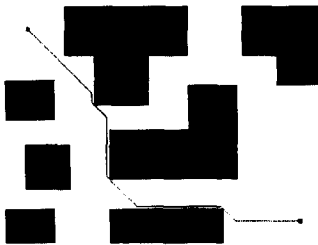


그림 8. DT

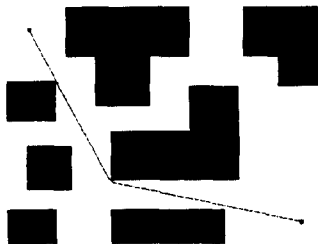


그림 9. Primitive DT

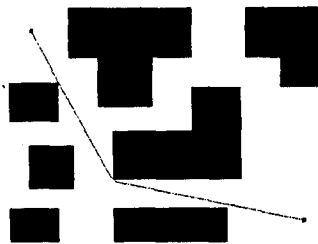


그림 10. Fast Primitive DT

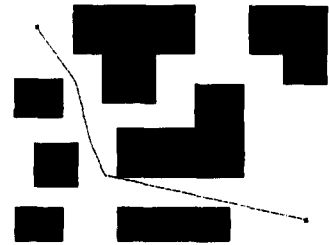


그림 11. SGA

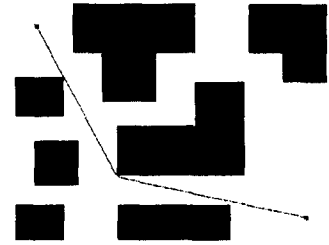


그림 12. Land Gambling DT

	시간(sec)	Length	Turn
DT	0.05	79.35	7
Primitive DT	7200	74.24	3
Fast Primitive DT	4.94	74.24	3
SGA	7.64	75.26	4
Land Gambling DT	1.37	74.24	2

표 2. 64×48 Map에서 실험 결과

### III. 참고문헌

- [1] A. Resonfeld and J.L. Pfaltz, "Sequential Operations in Digital Image Processing", *J.A.C.M.*, Vol. 13, No. 4, Oct, pp. 471-494, 1966.
- [2] R.A Jarvis, "Distance Transform Based Path Planning for Robot Navigation", *Recent Trends in Mobile Robots*, pp. 3-31, 1993.
- [3] Borenstein, J. and Koren, Y., "The Vector Field Histogram-Fast Obstacle-Avoidance for Mobile Robots", *IEEE Journal of Robotics and Automation*, Vol. 7, No. 3, June, pp 278-288, 1991.