

자율형 무인운반차를 위한 이동경로의 생성에 관한 연구  
- A Study on Moving Path Generation for Autonomous Vehicle -

임재국\*  
Lim, Jae-Kook  
이동형\*\*  
Lee, Dong-Hyung

ABSTRACT

This paper describes a moving path generation method for the Autonomous vehicles (AV) to search for paths in an unknown environment by using fixed obstacle information. Algorithms for the the AV which were recently proposed have some problems, so it was difficult to utilize these algorithms in the real world. The purpose of this research is to examine the applicability of real-time control and efficient improvement by reducing calculation iterations. In the network which is constructed by the cell-decomposition method, a gate is installed in each cell. By verifying the possibility of gate pass-over, the number of cells which should be considered to find the solution can be reduce. Therefore, algorithm iterations can be dramatically improved. In this paper we have proven that path-generated algorithms are efficient by using simluation.

1. 서론

소비자 니즈(Needs)의 다양화에 의해 필요한 것을 필요한 때에 만드는 다품종 소량생산시대가 도래함으로써 유연생산시스템의 도입이 활발히 진행중이다. 또한 차세대의 생산시스템으로서 변종변량생산을 전제로한 가공, 조립, 검사 등의 자율적인 모듈군으로 구성되는 초유연 생산시스템이 기대되고 있다. 이와같은 초유연생산시스템은 각 모듈자체가 유연해야 하고, 생산시스템에 대한 요구나 환경의 변화에 따라 모듈간의 조합이나 결합을 자율적으로 변경할 수 있는 자기조직화 기능을 갖게 될 것이다.

이러한 생산시스템의 변화추세에 따라 지금까지 대량생산을 전제로 해온 기존 공장의 설비의 배치나 운반기기 형태도 변화되고 있다. 산업혁명 이전에는 인력 중심의 운반이었고, 그후 대량생산시대가 도래되기전까지 간단한 동력이 부착된 운반기기가 사용되었다. 대량생산시대가 시작되는 20세기초에는 포드생산라인의 컨베이어나 배치생산과 같은 불연속적인 운반을 위한 포크리프트, 천정을 주행하는 크레인 등이 사용되었다. 또한 최근의 유연생산시스템(Flexible Manufacturing system ; FMS)에 있어서는 운반을 지원하는 무인운반차(Automatic Guided Vehicle ; AGV)가 등장하고 있다.

이러한 AGV는 무인운반 등 점점 복잡해져가는 환경 변화에 보다 잘 적응할 수 있도록 종래의 자동형(Automomic)에서 자율형(Autonomous)으로 발전되어 가고 있다. 따라서 자율모듈간을 연결하는 운반기기의 자율화를 위한 지속적인 연구가 필요하다.

\* 와세다 대학원 경영시스템공학과

\*\* 대전산업대학교 산업공학과

본 연구에서는 자율형 무인운반차(Autonomous Vehicle ; AV)가 목적지까지 이동하는데 있어 가장 중요한 기능인 경로계획, 즉 장애물(시설이나 작업장)의 위치정보를 얻어, 최단 이동경로를 자율적으로 생성하는 알고리즘을 제안한다. 또한 제안한 알고리즘을 복잡한 작업환경에 적용해 그 유효성을 검증한다.

## 2. 기존연구의 고찰

자율형 무인운반차의 이동경로생성문제(Autonomous Vehicle Problem)는 그림1에 표시한 것처럼 크게 정적 이동경로계획(static path planning)과 동적 이동경로계획(dynamic path planning)으로 나눌 수 있다.

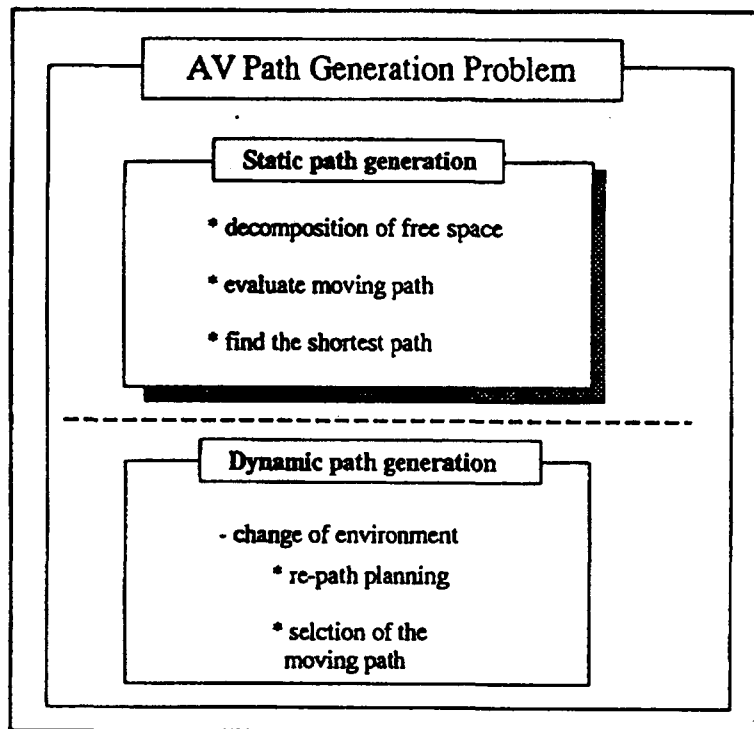


그림1. Research area

정적 이동경로계획은 AV가 이동을 시작하기전에 주어진 정보를 기초로 해 이동경로를 생성하는 것을 말한다. 다시 말해 환경정보가 주어진 그대로 변화없이 경로생성을 하는 경우이다. 이 경우에는 최초로 선정된 경로가 최후까지 변경되지 않는다.

동적경로계획은 환경정보가 시간경과( $\Delta t$ )에 따라 변화하는 경우의 경로생성을 말한다. 예를 들어 파렛트 등이 AV의 이동경로를 일시적으로 점유하고 있는 경우를 상정할 수 있다.

그동안 이동경로의 생성알고리즘에 대해서는 다양한 연구가 행해져 왔다. 우선, 작업장을 분할하는 일에 의해 구조화하고 이동경로를 생성하는 연구로서, Nilsson은작업장을 4×4의 셀(cell)로 분할하고, 셀내에 장애물이 있으면 다시 한번 4×4로 분할하고, 재차 장애물의 존재를 조사해 분할을 반복하는 방법을 제안했다[1]. 이 방법은 장애물의 형태를 확실하게 지시하지 않아도 모델이 구성되는 특징이 있다. 그러나 최단경로를 생성하기까지의 품이 들고, 계산량이 증가하는 문제가 있다.

Brooks는 작업장에 존재하는 장애물의 꼭지점(edge)에 대응되는 것을 선택하여, 그 중심선을 기준으로 뿔꼴모양의 중심단면과 같은 도형으로 작업장을 막아가는 수법을 제시하고 있다[2]. 그 수법은 기하학적인 간단한 형태로 작업장을 기술하는 것보다 AV의 통과여부의 체크 등이 용이하다. 그러나 이 수법은 최단경로를 발견하기 어렵다는 문제점을 해결하지 못하고 있다.

한편 작업장을 분할, 구조화 하지 않고 장애물에 관한 정보를 기초로 경로를 생성하는 방법이 Lozano-Perez에 의해 연구되어 왔다. 이는 작업장안에서 장애물을 오목 볼록 다각형으로 유사한 가시그래프를 구축해서 경로를 생성하는 수법이다[3]. 그러나 이 수법은 이동체의 크기를 과대하게 평가하고 있기 때문에 실제로는 통과가능한 지역도 불가능한 지역으로 평가되어지는 문제점이 있다. 곡선의 장애물까지 처리할수 있도록 Lozano-Perez의 수법을 확장한 연구도 발표되었다[4]. 이 수법은 장애물의 공통점선에 주목해서 그래프를 구성하는 것이지만, 이 수법의 문제점으로서 주어진 환경모델이 불완전한 경우에는 경로계획을 사전에 할 수 없고, 이동하면서 얻어진 환경정보를 이용해서 대차경로를 수정하지 않으면 안되는 일이 발생한다.

이상과 같이 기존 연구의 문제점을 정리하면 (1) 최단경로의 선택을 할 수 없는 경우 발생 (2) 경로의 생성에 방대한 계산시간 소요 라는 두가지지를 들 수 있다. 그러나 그동안 이러한 문제점을 해결하기 위해 다양한 연구가 행하여져 왔지만, 아직까지 확실한 해결책은 제시되지 않고 있다.

### 3. 제안하는 이동경로 생성법

정적이동 경로계획은 크게 공간의 분할문제, 통과여부의 체크, 최단이동경로의 생성알고리즘으로 구성된다. 본 연구는 이러한 정적이동 경로계획을 연구의 대상으로 하고 있으며 특히 위에서 언급한 문제점의 해소를 위해 게이트(gate)개념을 도입한 최단이동경로의 생성에 중점을 둔다. 이러한 게이트(gate)개념을 도입한 경로생성알고리즘은 다음과 같다. 또한 그림2는 본연구의 개요를 나타내고 있다.

#### < 이동경로생성 알고리즘 >

##### 제 1 단계 : 작업장의 분할과 게이트의 설치

- ① : 자유공간(free space)을 셀분할
- ② : 게이트의 설치

##### 제 2 단계 : 통과여부의 체크

- ① : 셀을 연결하는 연결그래프의 작성
- ② : 이동경로의 네트워크화
- ③ : 이동경로의 평가

##### 제 3 단계 : 최단이동경로의 생성

- ① : 이동경로의 계측
- ② : 최단이동경로의 생성

#### 3.1 작업장의 분할과 게이트의 설치

단계 1은 장애물의 꼭지점을 기준으로 해서 횡으로 선을 긋는 셀분할법에 의해 작업장을 분

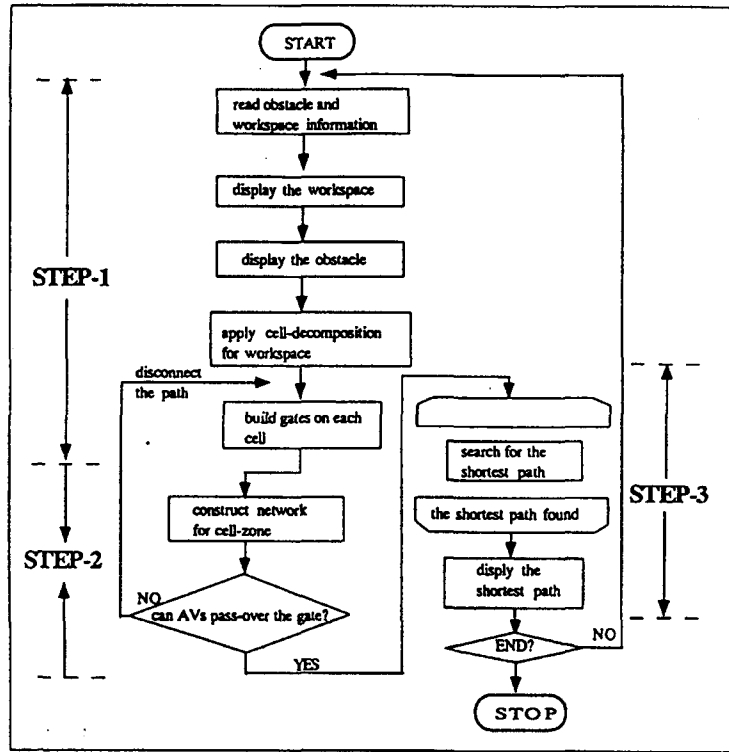


그림2. An algorithm of the proposed method

할하여, 생성된 셀에 게이트를 설치한다. 여기에서 셀은 「자유공간(free space ; 작업장)을 장애물의 꼭지점을 기준으로 해서 횡으로 선을 긋는 방법으로 분할하는 것에 의해 형성되어 벽이나 장애물, 게이트에 둘러싸인 공간의 부분집합」이라고 정의한다.

그림3과 같은 작업환경이 주어졌을 때 우선 그림4와 같이 자유공간을 분할한다. 그림4에서는 알파벳으로 표시된 사다리꼴 또는 장방형이 셀이다. 각 셀내의 임의의 2점을 연결하는

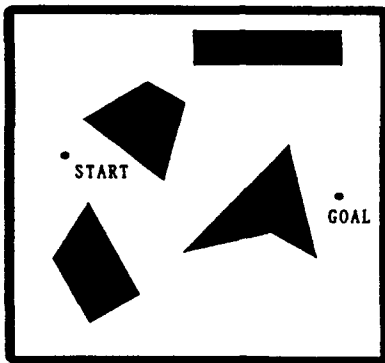


그림3. Free space

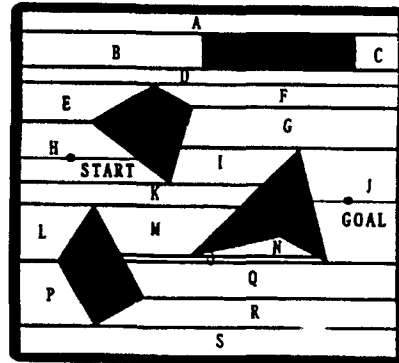


그림4. Slicing of free space by cell-decomposition

선분은 작업장내에 들어있기 때문에 그안에서는 경로가 용이하게 결정가능하다. 그림.4를 보면 얻어진 전체의 셀의 합집합이 자유공간을 정확하게 표시하고 있는 것을 알게 된다

다음은 생성된 셀에 게이트를 설치한다. 게이트란 “셀에 설치되어 셀의 정보를 관리하기 위한 표시” 라고 부여한다. 하나하나의 게이트에는 고유번호를 부여한다[5].

그림5는 셀에 설치하는 게이트의 설치방법을 나타내고 있다.

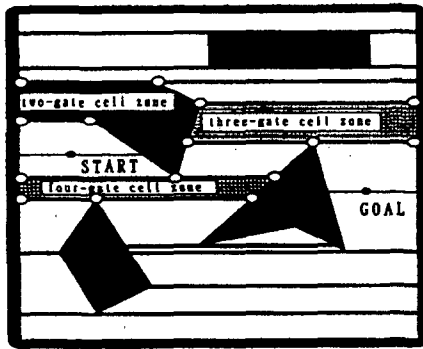


그림5. Build Gates on each Cell

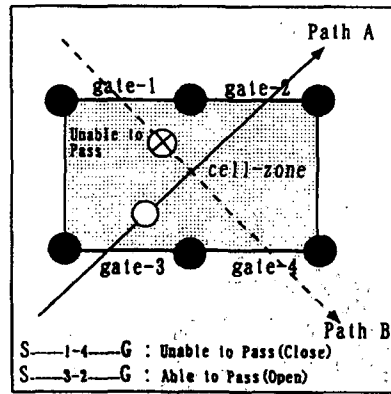


그림6. Management of gate information

그림6은 게이트가 세개인 경우의 예로 게이트정보를 이용한 셀의 정보를 관리하는 방법을 나타내고 있다. 경로 B의 경우는 셀안의 장애물과 장애물간의 간격을 측정해 AV가 그지역을 통과할 수 없다고 판정된 경우이다. 이 경우는 게이트 1-3으로 표시되는 루트를 막고, 경로생성대상에서 제외한다. 이 조작에 의해 탐색해야할 공간을 좁힐수 있고, 계산시간을 대폭 줄일 수 있다. 경로 A는 간격이 충분히 통과할 수 있다는 경우로 판정된 경우이다.

### 3.2 통과여부의 체크

단계 2는 셀을 연결하는 연결그래프를 구축하고, 이동경로를 네트워크화해서 통과여부를 체크한다. 다음은 그림7에 표시하고 있는 것과 같이 인접한 각 게이트상의 임의의 점 또는 START, GOAL점을 상호 연결하는 연결그래프를 작성한다. 연결그래프의 작성이 끝나면, START점에서 GOAL점까지의 이동경로의 후보를 네트워크화한다.

마지막으로 이동경로의 셀에 있는 장애물간의 간격과 AV의 크기를 비교하여 그림6의 이동 경로 B와 같이 통과할 수 없는 경우에는 루트를 폐쇄하고, 이동경로의 후보에서 탈락시킨다.

### 3.3 최단이동경로의 생성

단계 3은 남겨진 것 중에서 통과가능한 경로후보군에서 최단경로를 생성한다.

우선, 최단경로의 생성은 이동경로생성 모듈리[6]를 이용해서 이동경로의 거리를 측정한다. 이 모듈리의 특징은 장애물의 꼭지점에 AV의 크기를 고려한 안전지역의 설치와 실제이동한 거리를 기초로 하여 최단경로를 생성시키는 것이다. 특히, 이모듈의 이용에 의해 어떠한 형태의 장애물이라도 간섭하는 일 없이 이동경로의 생성을 할 수 있다. 즉, 이동경로는 고정장애물의 가장자리에서 안전거리+AV의 반경만큼 떨어져있다. AV는 이동가능한 경로의 후보가 결정되면, 이동경로의 형태를 인식해 모듈의 조합에 의해 최단경로를 생성한다.

### 3.3.1 최단 이동경로생성 모듈러

이동경로생성 모듈러는 그림8과 같이 4개의 부분으로 구성되어 있다.

서브모듈 1은 AV의 작업장 정보 및 초기 고정장애물 정보를 인식하는 “환경정보입력 및 장애물 인식부”이다. 서브모듈 2는 AV가 실제 이동하는 경로를 생성하는 “실용 이동경로 생성부”이다. 이곳에서는 장애물간의 간격을 측정해 그 통로를 AV가 통과할 수 있는가를 판단하고, 또한 지역내에 정체요인에 대한 정보를 분석 판단한다. 서브모듈 3은 “최단경로산출부”는 선정되어진 경로후보군으로부터 경로의 형상을 인식해 필요한 계측모듈을 조합해 이동경로의 거리를 정확하게 측정한다. 그 중에서 거리가 가장 짧은 경로를 초기 이동경로후보로 선정한다. 서브모듈 4는 시뮬레이션의 결과를 출력하는 “결과출력부”이다.

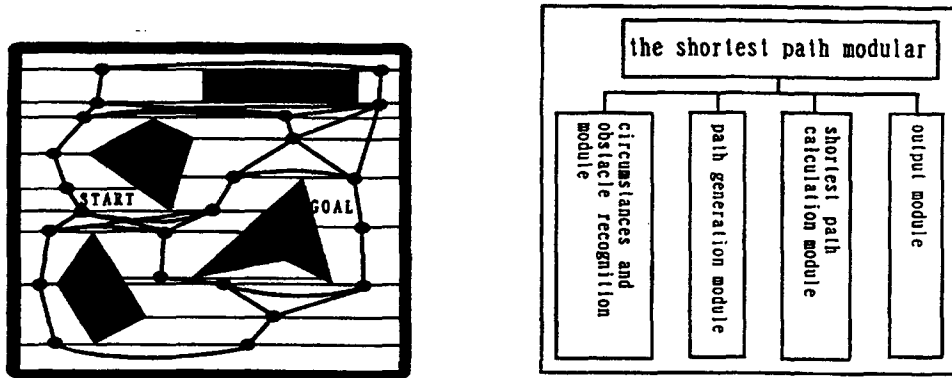


그림7. Composition of the shortest path modular    그림8. Composition of simulator

### 3.3.2 계측모듈의 구성

계측모듈은 그림9에 설명하고 있는 것과 같이 5가지의 서브모듈로 구성되어 있다. 그리고 각각의 계측모듈들은 다음과 같은 계산식에 의해 계산되어 진다.

$$\theta_1 = \tan^{-1} \frac{P'P_2}{P_1P'}$$

$$\theta_2 = \tan^{-1} \frac{P_3P'}{P_2P'}$$

$$\theta_3 = \cos^{-1} \frac{d}{P_1P_2}$$

$$\theta_4 = \cos^{-1} \frac{d}{P_2P_3}$$

$$\alpha = 360^\circ - (90 - \theta_1) - \theta_2 - \theta_3 - \theta_4 - 90^\circ$$

(START → GOAL)

$$\begin{aligned} \alpha' &= 180^\circ - \theta_1 + \theta_2 - \theta_3 - \theta_4 \\ &= 180 + \theta_1 - \theta_2 - \theta_3 - \theta_4 \end{aligned}$$

(GOAL → START)

$$\begin{aligned} \bar{d} &= \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \\ D_2 &= \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \\ DX_1 &= D_2 \times \sin\left(\cos^{-1}\left(\frac{d}{D_2}\right)\right) \dots\dots\dots (1) \end{aligned}$$

식(1)은 출발지점에서 도착지점으로 이동할 때, 장애물의 꼭지점을 기준으로하여 회전을 시작하기전까지를 나타내는 모듈공식이다.

$$\begin{aligned} D_2 &= \sqrt{(x_2 - x_3)^2 + (y_2 - y_3)^2} \\ DX_2 &= D_2 \times \cos\left(\sin^{-1}\left(\frac{d}{D_2}\right)\right) \dots\dots\dots (2) \end{aligned}$$

식(2)는 첫 장애물의 꼭지점을 돌아 다음 장애물의 꼭지점을 회전하기 시작하는 포인트까지의 거리를 나타내는 모듈공식이다.

$$DX_3 = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \dots\dots\dots (3)$$

식(3)은 장애물의 간섭없이 목표지점까지 이동이 가능할 때 사용되는 모듈공식이다.

$$\begin{aligned} D_2 &= \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \\ D_4 &= |x_2 - x_1| \\ D_6 &= \sqrt{(x_2 - x_3)^2 + (y_2 - y_3)^2} \\ D_8 &= |x_3 - x_2| \\ \alpha &= \left\{ 180 + \cos^{-1}\left(\frac{D_4}{D_2}\right) - \cos^{-1}\left(\frac{D_8}{D_6}\right) - \cos^{-1}\left(\frac{d}{D_2}\right) - \cos^{-1}\left(\frac{d}{D_6}\right) \right\} \times \frac{3.14}{180} \\ DX_4 &= d \times \alpha \dots\dots\dots (4) \end{aligned}$$

식(4)는 장애물의 꼭지점을 돌아갈 때 회전하여 다음 포인트로 이동하기 전에 행하는 회전 거리를 나타내는 모듈공식이다.

$$\begin{aligned} D_2 &= \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \\ D_4 &= |x_2 - x_1| \\ D_6 &= \sqrt{(x_2 - x_3)^2 + (y_2 - y_3)^2} \\ D_8 &= |x_3 - x_2| \\ \alpha &= \left\{ 180 - \cos^{-1}\left(\frac{D_4}{D_2}\right) + \cos^{-1}\left(\frac{D_8}{D_6}\right) - \cos^{-1}\left(\frac{d}{D_2}\right) - \cos^{-1}\left(\frac{d}{D_6}\right) \right\} \times \frac{3.14}{180} \\ DX_5 &= d \times \alpha \dots\dots\dots (5) \end{aligned}$$

식(5)는 식(4)와 반대 방향으로 복귀하는 것을 고려할때 사용되는 회전거리 모듈공식이다.

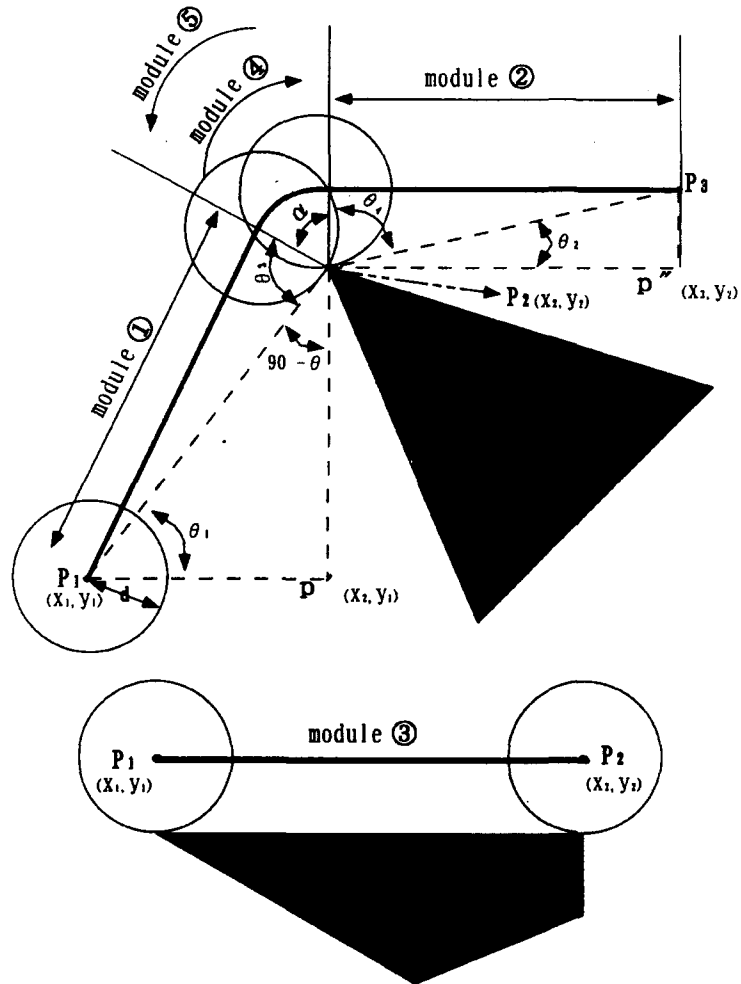


그림9. Composition of measurement module

#### 4. 사례에 의한 검증

이상 제안한 알고리즘을 복잡한 작업환경을 갖는 사례에 적용하여 그 유효성을 검증한다

##### 4.1 시뮬레이션의 조건

시뮬레이션은 이하의 조건에 의해 행하여졌다.

- 작업장의 사이즈는 600×600으로 한다.
- 장애물은 복잡한 형태를 갖고 있다. (그림9 참조)
- AV의 폭은 30으로 한다.
- 경로의 생성은 AV 1대가 행하는 것으로 한다.
- 초기 고정장애물의 정보를 갖고 있다.



4.2 시뮬레이션 결과

이상의 조건에 의해 시뮬레이션을 행하여 다음과 같은 결과를 얻었다.

그림10의 (a)는 작업장을 셀분해하여 생성된 셀에 게이트를 설치한 결과, 18개의 게이트가 설치되어있는 것을 나타내고 있다. 여기에서는 통과불가능한 지역으로 한군데가 체크되었다. (☆표시) 또한 이동경로 네트워크에서 통과불가능한 셀을 통과하는 경로네트워크를 제외한 결과, flag "1"에 표시된 루트 2,3,5와 루트 6이 이동경로 후보로 남았다.

그림10의 (b)는 남은 4개의 루트의 거리를 이동경로 생성모듈에 의해 계측한 결과, 최단이동경로로 루트 6이 경로로서 선정되어진 것을 보여주고 있다.

그림10의 (c)는 시뮬레이터의 결과화면이다. 여기에서의 R-num은 이동경로의 네트워크번호를 표시하고 있다. 여기에서는 6개의 이동경로 네트워크가 생성되어진 것을 보여주고 있고, flag1은 통과가능, 0은 통과불가능한 것을 나타내고 있다. 이 판정은 본 논문에서 제안한 게이트의 개념에 의한다. 마지막으로 DOOR는 이동경로네트워크를 연결하는 게이트의 번호를 표시하고 있다.

```

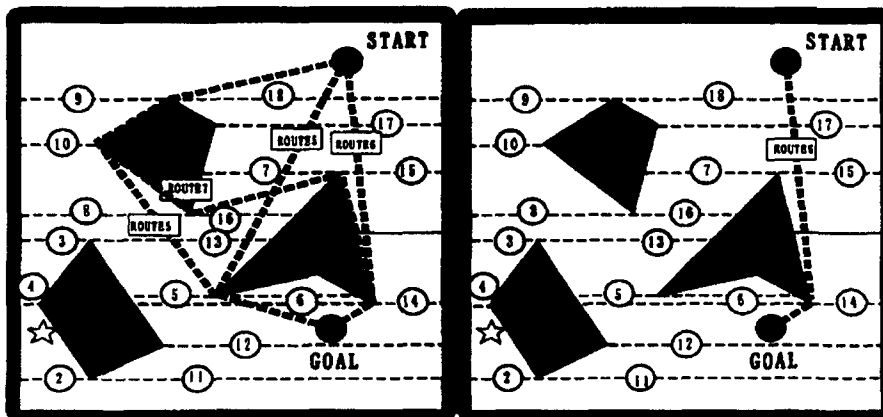
----> ROUTE and STATUS <-----
R_num | flag | DOOR
  1    | 0    | 1-9-10-8-3-4-2-11-12-1000
  2    | 1    | 1-9-10-8-13-5-6-1000
  3    | 1    | 1-9-10-8-16-7-15-14-1000
  4    | 0    | 1-18-17-7-16-3-4-2-11-12-1000
  5    | 1    | 1-18-17-7-16-13-5-6-1000
  6    | 1    | 1-18-17-15-14-1000
----> ROUTE and STATUS <-----
R_num | flag | DOOR
  2    | 1    | 1-9-10-8-13-5-6-1000
  3    | 1    | 1-9-10-8-16-7-15-14-1000
  5    | 1    | 1-18-17-7-16-13-5-6-1000
  6    | 1    | 1-18-17-15-14-1000

----> The result of searching process. <-----
-> Route 2 : distance is 1108.59.
-> Route 3 : distance is 1401.35.
-> Route 5 : distance is 565.88.
-> Route 6 : distance is 493.36.

-> The Shortest Route is Route-NUM 6.

-----> End of Program <-----
    
```

(C)



(a) (b)  
그림10. Example of the shortest path be penetrated

## 5. 결론

본 연구에서는 게이트 개념을 이용해서 AV의 경로계획에 있는 정적인 이동경로를 생성하는 알고리즘을 제안했다. 제안한 알고리즘에 의해 이동경로의 생성을 보다 간단한 문제로 귀착시켰다. 즉, 장애물과 장애물간의 폭, 즉 게이트의 폭의 검토에 의해 통과할 수 없는 루트를 폐쇄하여 이동경로의 생성대상에서 제외하는 것에 의해 탐색해야 할 해공간을 좁힐수 있다. 또한 복잡한 작업장 환경을 상정한 사례에 의한 검증을 행한 결과, 통과가능한 이동경로 후보군에서 최단이동경로가 생성되는 일이 보증할수 있게 되어 그 유효성이 증명되었다.

따라서 이 연구결과 종래 경로탐색문제에 있어 커다란 과제로 남아 있던 계산량 문제가 해결가능하게 되었으며 AV의 실시간 제어의 가능성을 높였다.

## 참고 문헌

- [1] Nilsson N., *Principles of artificial intelligence*, Tioga publishing, 1980.
- [2] Brooks R. A., Solving find-path problem by good representation of free space, *IEEE J. of Trans. on SMC*, SMC-13, pp.190-197, 1983.
- [3] Lozano-perez T. and Wesley M.A., An algorithm for planning collision - free paths among polyhedral obstacles, *Communication of the ACM*, Vol. 30, No.10, pp.560-570, 1979.
- [4] Y. H. Liu and S. Arimoto., A flexible algorithm for planning local shortest path of mobile robots based on reachability graph, *Proc. IEEE International Workshop on Intelligent Robots and system*, 1990.
- [5] Jae-Kook Lim and T. Takahashi, An approach to dynamic path generation based on cell-decomposition, *JDME The 4th Transportation and Logistics conference*, 1995.
- [6] Jae-Kook Lim and T. Takahashi, An approach to reconstruction of moving path based on reconnaissance simulator, *JSME The 5th Transportation and Logistics conference*, 1996.