

# Flow Network 을 이용한 청소로봇의 최소방향전환 경로계획

## Minimal Turning Path Planning for Cleaning Robots Employing Flow Networks

남 상 현, 문 승 빈\*  
(Sang-Hyun Nam and Seungbin Moon)

**Abstract :** This paper describes an algorithm for minimal turning complete coverage path planning for cleaning robots. This algorithm divides the whole cleaning area by cellular decomposition, and then provides the path planning among the cells employing a flow network. It also provides specific path planning inside each cell guaranteeing the minimal turning of the robots. The minimal turning of the robots is directly related to the faster motion and energy saving. The proposed algorithm is compared with previous approaches in simulation and the result shows the validity of the algorithm.

**Keywords :** complete coverage path planning, template, cellular decomposition, cleaning robot, mobile robots

### 1. 서론

최근 자율 이동로봇의 경로 계획에 관한 연구가 활발히 진행되고 있다. 이동로봇에 대한 경로 계획은 크게 PTP방식(Point-to-Point)과 [1,2] 전체영역 경로계획(CCPP, Complete Coverage Path Planning)방식 [3-13]으로 분류된다. PTP방식이란 우편배달 로봇과 같이 주어진 두 점 사이를 최적의 경로를 이루면서 이동하는 방법을 말하며, CCPP방식은 청소로봇 또는 지뢰탐사로봇과 같이 전체 영역을 이동하여야 하는 경우에 적용되는 방법을 말한다.

전체영역 경로계획은 대표적으로 템플릿(template)을 이용한 방식과 [3,4,5], potential field를 이용한 방식 [6,7], 셀 분할을 이용한 방식으로 [8-12] 분류할 수 있다. 템플릿을 이용한 방식은 몇 개의 기본 경로 계획 템플릿과 청소 환경을 이용하여 경로계획을 한다. 로봇과 환경 지도를 기반으로 가장 적당한 템플릿을 이용한다. Hofner등은 빠른 방향전환을 위하여 여러 모양의 템플릿으로 경로계획을 하였으며 [3], 또한 청소를 실시한 후에 아직 청소가 안된 영역에 대한 재 경로 계획방법도 제안하였다[4]. Carvalho등은 변화된 장애물이 있는 경우에도, 로봇이 기본 경로로 전환 될 때까지 예상치 못한 장애물의 윤곽을 단순히 따라가 경로를 생성하는 방법이 제안되기도 하였다[5]. Nam등은 실제 청소환경에서 비전을 이용한 장애물 인식과 자기위치를 추정하는 방식을 연구하였다[13].

Potential field를 이용한 방식에서는 청소 영역을 수치적 값을 가진 셀로 표현한다. 방문한 셀의 값은 증가하고 주위에서 가장 낮은 값을 가진 셀로 이동하는 경로 계획을 말한다. 이때 발생하는 많은 방향전환을 줄이기 위하여 셀의 값으로 거리함수와 방향전환 함수를 이용하였다[6]. Oh등은 삼각형 셀을 이용하여 청소영역을 표현하는 포텐셜 필드 방식을 제안하였다[7].

셀 분할을 이용한 방식은 청소공간을 셀로 나뉘서 해석하

는 경로계획을 말한다[8-12]. Gabriely등은 경로를 결정하기 위해 현재의 셀에서 주변의 셀로 우회하면서 확장해나가는 STC(Spanning-Tree Coverage)방식을 제안하였다[8].

Choset은 장애물을 만나는 임계점에서 셀들을 분리하고, 장애물을 벗어나는 임계점에서 나뉘어진 셀들을 합치는 boustrophedon 셀 분할방식을 제안하였다[10]. Acar등은 셀 영역을 덮기 위해 단순히 back-and-forth motions을 이용하였고 셀 영역을 그래프로 나타내어 단순한 그래프 탐색 알고리즘인 깊이 우선 탐색방식을 이용해 전체 영역을 방문하는 morse decomposition 방식을 도입하였다[12]. 또한, 실제의 환경에서 경로계획을 하기 위해 청소영역을 4가지 임계점으로 표현하는 방법[9]과 알 수 없는 환경에서의 경로계획방법을 연구하였다[11].

자율 청소로봇의 경우 많은 사용자들은 전체청소 경로계획을 이용해 좀더 빠르고 효율적으로 청소하기를 원한다. 좀더 빠르고 효율적으로 청소를 하기 위해서는 로봇의 방향전환을 최소한으로 줄여야만 한다. 대부분 방향전환을 하기 위해서는 감속을 하면서 로봇을 정지시키고 다시 출발하기 위하여 가속을 반복하게 된다. 그 결과 많은 시간이 소비되고, 또한 로봇이 방향전환을 위하여 정지를 해야 됨으로 가/감속을 하면서 많은 에너지 소비를 하게 된다. 따라서 청소영역을 더 효과적으로 이동하기 위해서는 청소영역의 모양에 따라 최소의 방향전환을 갖는 경로계획이 필요하다.

본 논문에서는 청소영역의 효과적인 경로 생성을 위하여 셀 분할방식에 플로우네트워크(flow network)를 접목한 방식을 제안하였다. 넓은 청소공간을 여러 개의 플로우네트워크로 표현함으로써 좀더 쉽게 경로계획을 할 수 있다. 본 알고리즘은 전체 청소영역을 셀들로 분리하고 셀들을 플로우네트워크로 표현하고 셀들 사이의 경로를 최적화하였다. 그리고 각 셀에 입출력 부분정점을 설계해서 각 셀의 모양에 따라 셀에 맞는 최적의 경로들을 생성해주는 전 영역 경로계획을 작성하였다. 본 논문의 2절에서는 임계점을 기준으로 청소영역을 셀들로 나누는 셀 분할방식과 경로계획에 따른 방향전환 횟수에 대하여 살펴 보았다. 그리고 3절에서는 플로우네트워크를 이용한 전체영역 경로계획과 최소 방향전환을 위한

\* 책임저자(Corresponding Author)

논문접수 : 2004. 12. 14., 채택확정 : 2005. 7. 1.

남상현, 문승빈 : 세종대학교 컴퓨터공학과

(shnam28@robotics.sejong.ac.kr/sbmoon@sejong.ac.kr)

셀 내부 경로계획에 대해 세부적으로 설명하였고 알 수 없는 환경에서 임계점을 찾는 순환알고리즘을 제안하였다. 그리고 4절에서는 기존의 방식과 제안된 방식의 전 영역경로계획을 시뮬레이션으로 비교하였고, 마지막으로 5절에서 결론을 도출하였다.

**II. 셀 분할 방식**

**1. 셀 분할 방식**

그림 1에서 보는 바와 같이, 청소영역은 장애물의 임계점(critical points)에 따라서 여러 개의 셀, Cell<sub>i</sub>, 로 나타낸다. 여기서 임계점은 장애물의 진입부와 마지막 부분에 있는 돌출부분을 나타내는 것으로[9], 그림 1에서는 CP<sub>1</sub>로 표시하였다. 그림 1에서 청소영역은, CP<sub>1</sub>을 기준으로 Cell<sub>1</sub>과 Cell<sub>2</sub>의 두 개의 셀로 분리되며, CP<sub>2</sub>를 벗어나면 한 개의 셀 Cell<sub>3</sub>로 합쳐지게 된다. 그리고 CP<sub>3</sub>을 기준으로 Cell<sub>4</sub>와 Cell<sub>5</sub>의 두 개의 셀로 다시 분리된다. 이때 Cell<sub>4</sub>는 오목한 모양의 출구가 막혀 있는 형태를 가지게 된다. 그리고 CP<sub>4</sub>를 기준으로 다시 Cell<sub>6</sub>으로 합쳐지게 된다. 이렇게 분리된 각 셀의 중심은 무게중심 P<sub>i</sub>로 그림 1에 나타내었다.

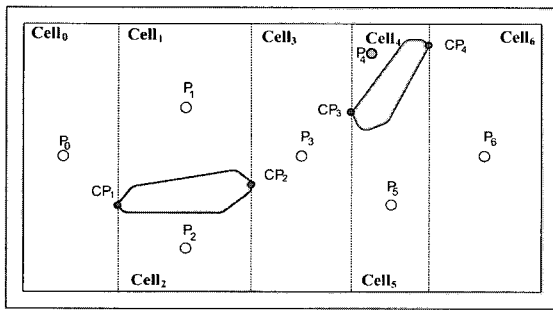
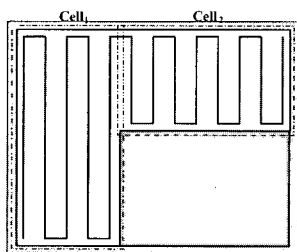
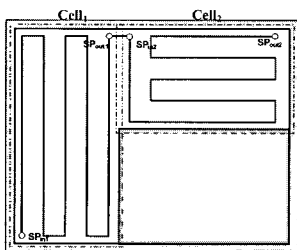


그림 1. 셀 분할방법을 이용한 청소 영역 구분.  
Fig. 1. Partition of cleaning area by cellular decomposition.



(a) when directions are identical



(b) when directions are different

그림 2. 진행방향 변화에 따른 방향전환수의 비교.  
Fig. 2. Comparison of the number of turnings.

**2. 경로계획에 따른 방향전환 수의 비교**

공간을 빠르게 청소하면서 에너지 효율을 높이기 위해서는 로봇의 방향전환을 최소한으로 줄여야만 한다. 일반적으로 로봇이 방향전환을 할 때 정지를 하기 때문에 많은 시간이 소비되고, 또한 정지 및 출발 시에 가/감속을 하여야 함으로 많은 에너지가 낭비된다.

따라서, 방향전환 수를 줄이기 위하여 청소영역의 모양에 따라 최소의 방향전환이 발생하도록 청소 진행 방향을 결정할 필요가 있다. 예를 들어, 그림 2(a)에서는 Cell1과 Cell2의 주경로가 동일하며, 이에 비하여 그림 2(b)에서는 Cell1과 Cell2의 주경로 방향이 다르게 되었음을 볼 수 있다. 이러한 경우에, 그림 2(a)에서는 Cell2를 청소하기 위해 15회의 방향전환이 필요한데 비하여, 그림 2(b)에서는 Cell2를 청소하기 위해 10회의 방향전환만이 필요한 것을 볼 수 있다. 방향 전환수를 줄이기 위해서는 로봇의 주 이동경로를 셀의 형태에 따라 변경하는 것이 필요하다. 일반적으로, 셀의 길이가 긴 방향으로 로봇의 주경로를 설정하게 되면 로봇의 방향전환 횟수가 줄어들게 된다.

**III. 플로우네트워크용 최소방향전환 알고리즘**

**1. 플로우네트워크**

그림 3은, 그림 1에서 보여주었던 분할된 청소영역을 왼쪽에서 오른쪽으로 흐름을 갖는 플로우네트워크로 나타내기 위하여, 각각의 셀을 중심점 P<sub>i</sub>에서의 정점(node)으로 표현하고, 각 셀들 사이의 이동 경로는 간선(branch)으로 표현한 것이다. 정점은 플로우네트워크가 시작되는 시작정점과 마지막인 도착정점, 그리고 그 외의 중간정점으로 구분한다. 간선은 단방향 간선과 양방향 간선이 있는데, 단방향 간선이란 화살표로 표시된 방향으로만 이동이 가능한 간선을 의미하며, 양방향 간선이란 양쪽 방향으로 모두 이동이 가능한 간선을 말한다.

시작정점은 인접한 정점까지 오른쪽 방향으로 향하는 단방향 간선을 갖는다. 마찬가지로, 도착정점도 인접한 정점으로부터 도착정점으로 입력되는 단방향 간선을 갖게 된다. 시작정점과 도착정점을 제외한 중간정점들은 양방향 간선을 갖게 된다. 여기서, 양방향 간선은 단 방향 간선보다 경로계획 시 우선순위를 높게 함으로서, 도착정점에 도달하기 전에 양방향 간선과 연결된 모든 중간정점을 거치도록 한다.

그림 3에서, 양방향간선은 양방향 화살표로, 단방향 간선은 단방향 화살표로 표현하였으며, 각각의 간선은 두 정점

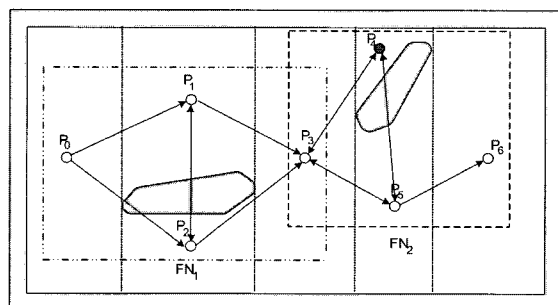


그림 3. 청소영역을 플로우네트워크로 표현.  
Fig. 3. Flow network for cleaning area.

사이의 거리에 해당하는 가중치를 갖도록 하였다. 그림 3의 플로우네트웍은  $FN_1 = \{P_0, P_1, P_2, P_3\}$ 과  $FN_2 = \{P_3, P_4, P_5, P_6\}$  두 개의 플로우네트웍으로 구성이 된 것을 볼 수 있다.  $FN_1$ 에서는  $P_0$ 가 시작정점이 되고  $P_3$ 가 도착정점이 되며,  $FN_2$ 에서는 시작정점은  $P_3$ 가 되고 도착정점은  $P_6$ 가 된다. 여기서,  $P_3$ 는  $FN_1$ 의 도착 정점과 동시에  $FN_2$ 의 시작정점이 됨을 알 수 있다.

또한,  $FN_2$ 에서는 3면이 막힌 셀의 중간정점인  $P_4$ 가 존재하는데 이러한 중간정점을 특별히 흑색 정점으로 나타내었다.  $P_4$ 와 같이 3면이 막힌 지역의 정점은 시작정점과 인접해 있어도 단방향 간선을 갖지 않고 양방향 간선을 갖게 함으로서, 다른 정점보다 먼저 시작정점과 연결되도록 한다. 그 이유는 이 점이 도착정점과는 장애물로 막혀있어서 연결이 될 수 없으므로 가급적 도착정점과 연결되는 정점이 되지 않도록 고려한 것이다.

2. 셀간의 경로 계획

전체영역 경로계획을 완성하기 위하여, 첫째로 각셀간의 경로계획을 하여야 한다. 이를 위하여, 그림 3에서와 같이 두 개의 플로우네트웍,  $FN_1$ 과  $FN_2$ 에 대하여 Dijkstra 알고리즘을 [14] 이용하여 경로 계획을 작성한다. Dijkstra 알고리즘은 처음 정점에서 인접한 정점까지 해당 간선의 거리 가중치를 부여하고 거리 가중치가 가장 적은 정점으로 이동하는 방식을 말한다.  $FN_1$ 의  $P_0$ 부터  $P_3$ 까지 생성된 경로의 순서는  $Path_1 = \{P_0, P_1, P_2, P_3\}$ 이고 도착 정점인  $P_3$ 에 도착했을 때  $FN_1$ 은 완성된다.

$FN_2$ 의 경로계획은  $P_3$ 부터  $P_6$ 까지 경로를 생성하게 된다. 여기서 정점  $P_3$ 은  $P_4$ 와  $P_5$ 중 경로를 생성하게 되는데 3면이 닫힌 오목한 모양의 셀을 나타내는  $P_4$ 가 양방향 경로를 갖고 있으므로  $P_4$ 의 양방향 경로를 먼저 생성 후  $P_5$ 로 이동하게 된다. 즉,  $FN_2$ 의 경로 순서는  $Path_2 = \{P_3, P_4, P_5, P_6\}$ 이다. 2개의 플로우네트웍 사이의 경로계획을 그림 4에 나타내었다.

3. 최소 방향전환 셀 내부 경로 계획

플로우네트웍을 이용하여 셀간의 경로계획이 끝나면 셀 내부 경로계획이 필요하다. 각 셀들은 외각에 4개의 부분정점(sub-point)이 존재하게 되는데, 서로 인접한 두 셀의 부분정점 중에서 경로 시작점이 되는 입력 부분정점인  $SP_{in}$ 과 경로의 최종점이 되는 출력 부분정점인  $SP_{out}$ 을 구한다. 예를 들어, 그림 4에서  $Cell_0$ 에서  $Cell_1$ 로 이동경로가 생성된 것을 볼 수 있다.  $Cell_0$ 로부터  $Cell_1$ 까지 경로가 생성되면  $Cell_0$ 의 부분정점과  $Cell_1$ 의 부분정점 사이에 가장 인접한 2개의 부분정점이 존재하게 되는데 이때  $Cell_0$ 에 속한 부분정점은  $SP_{out}$ 이 되고  $Cell_1$ 에 속한 부분정점은  $SP_{in}$ 이 된다. 각 셀마다의  $SP_{in}$ 과  $SP_{out}$ 을 그림 4에 나타내었다.

2.2에서 설명한 바와 같이 셀 내부에서의 경로계획은 방향 전환 횟수를 최소화하기 위하여  $SP_{in}$ 과  $SP_{out}$ 의 위치 및 셀의 방향에 따라 주 경로가 긴 방향으로 하는 것이 유리하다. 그림 5는 부분정점의 모양에 따라 6가지 청소패턴을 나타낸 것이다. 여기서, 그림 5(d),(e),(f)는 각각 그림 5(a),(b),(c)와 셀의 방향만 틀릴 뿐 청소형태는 동일함을 알 수 있다. 따라서 그림 5(a),(b),(c)의 3가지 형태만을 고려하면 충분하며, 그 외의 경우는 이 3가지와 유사하게 구할 수 있음을 알 수 있다.

그림 5는 셀의 4면 중 2면은 장애물이나 벽으로 닫혀있고

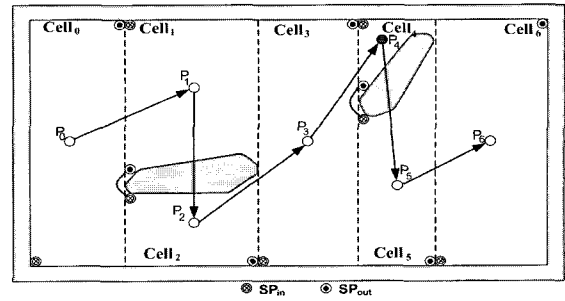


그림 4. 플로우네트웍을 이용한 셀간 경로계획.  
Fig. 4. Path planning employing flow network.

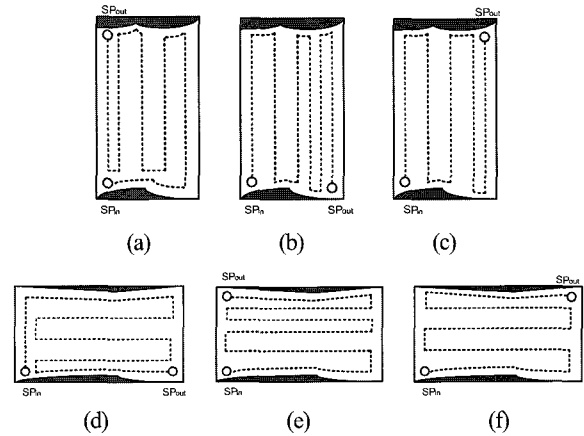


그림 5.  $SP_{in}$ 과  $SP_{out}$ 의 위치에 따른 경로계획.  
Fig. 5. Path planning with respect to positions of  $SP_{in}$  and  $SP_{out}$ .

나머지 2면은 개방되어 있는 경우를 나타내었다. 그리고 3면이 닫혀있는 오목한 형태의 지역은 그림 5(a) 또는 (e)의 경로계획을 하면 된다. 그리고 셀 내부의 경로는 벽이나 장애물 주변에서는 벽추종(Wall Following) 경로를 생성한다. 그림 5(a)의 셀을 보면 좌측 하단부에  $SP_{in}$ 이 존재하고 좌측 상단부에  $SP_{out}$ 이 존재한다. 여기서, 직전의 셀이 좌측 또는 아래에서 현재의 셀로 이동하였으며, 다음에 이동할 셀은 좌측 또는 위에 존재한다는 것을 예측할 수 있다. 그림 5(f)의 셀을 보면 직전의 셀이 좌측 또는 아래에서 현재의 셀로 이동하였고 다음에 이동할 셀은 우측 또는 위에 존재한다는 것을 예측할 수 있다. 그리고 각 셀과 셀 사이 이동 시  $SP_{in}$ 에서  $SP_{out}$ 방향으로 이동경로가 생성됨으로 전 영역 경로계획이 완성된다.

4. 전체영역 경로계획

3.2절에서의 셀간 경로계획과 3.3절에서의 셀 내부의 경로계획을 통합하면 그림 6과 같이 전영역 경로계획을 구할 수 있다. 각 셀마다의 경로계획을 살펴보면 그림 6에서  $Cell_0$ 은 그림 5(c) 형태의 경로계획임을 알 수 있고  $Cell_1$ 은 그림 5(a) 형태이고  $Cell_2$ 는 그림 5(f) 형태임을 알 수 있다. 그리고  $Cell_3$ ,  $Cell_5$ ,  $Cell_6$ 는 그림 5(c) 형태의 경로계획임을 알 수 있다.  $Cell_4$ 는 3면이 닫혀있는 오목한 지역이므로 그림 5(a) 형태의 경로계획을 이용하였음을 알 수 있다. 그리고 각 셀마다 생성된 경로들은 각 셀의  $SP_{out}$ 과  $SP_{in}$ 사이를 연결하는 경로를 생성함으로써 서로 연결될 수 있다.

5. 경로계획 수행 시 문제점과 해결방안

청소로봇이 실제로 주행 시 바퀴 미끄러짐 등에 의하여 위치예러 발생으로 인한 문제는 정확한 위치보정을 위하여 해결 되어야 한다. 이러한 위치보정에는 레이저 스캐너를 사용하는 방식[15]과 비전 카메라를 이용하는 방식 등을 채택할 수 있는데 우리는 위치예러를 보정하기 위해 비전카메라와 마크를 이용한 절대 위치와 각을 찾는 방법을 이용해 경로 계획 시 누적된 위치추정 오차를 보정하였다[13].

또한 장애물의 정보를 알 수 없는 환경에서 제안 된 경로 계획을 하기 위해선 장애물의 입력과 출력 임계점 정보를 알아야 한다. 이러한 임계점 정보를 알기 위해 순환알고리즘(cycle algorithm)을 제안하였다. 로봇이 기본경로로 이동 시 측면 센서로 장애물을 감지하고 장애물을 기준으로 두 개의 상위셀(a top cell)과 하위셀(a bottom cell)영역으로 분리된다고 가정하면, 이때 처음 시작되는 영역이 상위 셀 영역이 되고 남은 부분이 하위 셀 영역이 된다. 순환알고리즘의 동작은 두 단계로 구성되는데 그림 7에 FORWARD단계(a-b-c, e-f-g)는 점선으로 표시하였고 REVERSE단계(c-d-a, g-h-e)는 검정색 실선으로 표시하였다. 상위 셀에서 FORWARD단계(a-b-c)를 이용해 CP<sub>3</sub>, CP<sub>4</sub> 임계점을 찾았고 REVERSE단계(c-d-a)로 이동 시 감지된 장애물을 추종하여 CP<sub>1</sub>, CP<sub>2</sub> 임계점을 찾아 순환알고리즘이 완성하였다. 상위 셀에서 셀 내부의 경로계획이 완성되면 하위 셀로 이동하여 다시 순환알고리즘을 만든다. 이때 하위 셀의 순환알고리즘의 순서는 (e-f-g-h-e)로 상위 셀의 순환알고리즘 순서(a-b-c-d-a)와는 다르게 진행된다. 순환알고리즘으로 찾은 입력 출력 임계점을 이용해 청소영역은 셀로 분할 하게 된다.

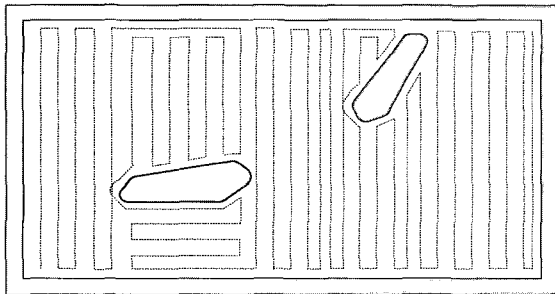


그림 6. 제안된 전체영역 경로계획방식.  
Fig. 6. Proposed complete coverage path planning.

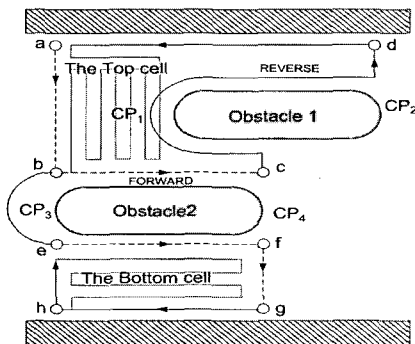


그림 7. 순환알고리즘을 이용한 임계점 찾는 방법.  
Fig. 7. Critical point search by circular algorithm.

IV. 시뮬레이션

본 논문에서 제안한 최소 방향전환 경로계획의 장점을 보이기 위하여, 기존에 Choset이 제안한 방식[12]에 의한 경로 계획과 제안된 알고리즘에 의한 경로계획을 비교하여 보았다. 그림 8은 장애물의 임계점을 기준으로 셀 영역을 분리하였다. 그리고 임계점은 점점으로 표현하고 셀들은 간선으로 표현하는 그래프를 이용하였다.

셀 영역의 경로계획은 깊이 우선탐색방식을 이용하여, 아래쪽의 셀을 끝까지 경로계획 한 후 뒤로 돌아오면서 경로계획이 안된 셀 영역을 경로계획 하는 것을 볼 수 있다. 그리고 셀과 셀 사이를 이동할 때는 정점인 임계점을 이용해서 이동한다. 그림 8을 보면 셀 영역의 경로 순서는 다음과 같다. Path = {Cell<sub>0</sub>, Cell<sub>2</sub>, Cell<sub>3</sub>, Cell<sub>5</sub>, Cell<sub>6</sub>, Cell<sub>4</sub>, Cell<sub>1</sub>}. 우선 아래에 위치한 Cell<sub>0</sub>, Cell<sub>2</sub>, Cell<sub>3</sub>, Cell<sub>5</sub>, Cell<sub>6</sub>을 경로계획 후 경로계획이 안된 Cell<sub>4</sub>, Cell<sub>1</sub>을 경로계획하였다. 그리고 경로계획이 안된 셀들 이동 시 임계점들을 이용해 Cell<sub>6</sub>의 마지막 정점에서 CP<sub>4</sub>로 이동하고 CP<sub>3</sub>에서 CP<sub>2</sub>로 이동됨을 볼 수 있다. 그림 9은 본 논문에서 제안한 플로우네트웍을 이용한 셀 분할 방식으로, 셀의 중심에 정점을 갖고 셀과 셀 사이의 경로를 간선으로 나타내었다. 셀들 사이의 경로계획은 2개의 플로우네트웍을 이용하여 경로계획을 하였다. 그림 9을 보면 전체 청소영역은 FN<sub>1</sub> = {Cell<sub>0</sub>, Cell<sub>1</sub>, Cell<sub>2</sub>, Cell<sub>3</sub>}과 FN<sub>2</sub> = {Cell<sub>3</sub>, Cell<sub>4</sub>, Cell<sub>5</sub>, Cell<sub>6</sub>}로 구성되었고 전체 경로순서는 다음과 같다. Path = {Cell<sub>0</sub>, Cell<sub>1</sub>, Cell<sub>2</sub>, Cell<sub>3</sub>, Cell<sub>5</sub>, Cell<sub>4</sub>, Cell<sub>6</sub>}. 전체 셀들의 경로 순서가 정해지면 각 셀 내부의 입출력 정점인 SP<sub>in</sub>과 SP<sub>out</sub>을 이용해 셀의 모양이 고려된 방향전환횟수가 적은 셀 내부의 경로 계획이 완성된다.

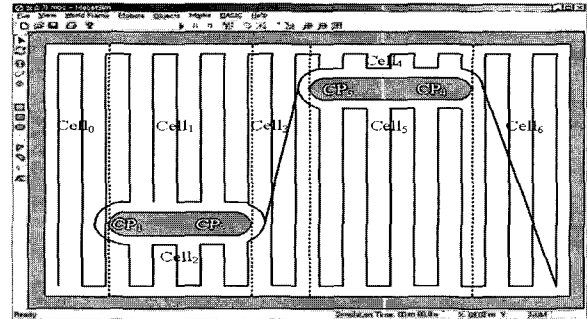


그림 8. 기존의 전체영역 경로계획방식.  
Fig. 8. Normal path planning.

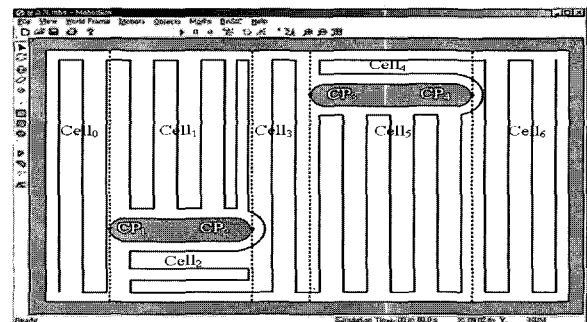


그림 9. 최소 방향전환 전체영역 경로계획.  
Fig. 9. Minimal turn path planning.

표 1. 시뮬레이션 결과.

Table 1. Simulation result.

	Total turns	Total time
Proposed Algorithm	54	556[sec]
Algorithm[12]	74	665[sec]

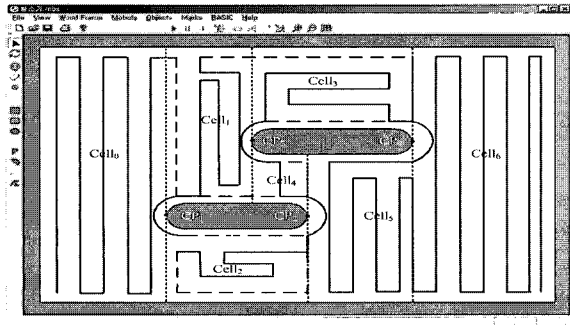


그림 10. 순환알고리즘을 이용한 경로계획.

Fig. 10. Path planning employing circular algorithm.

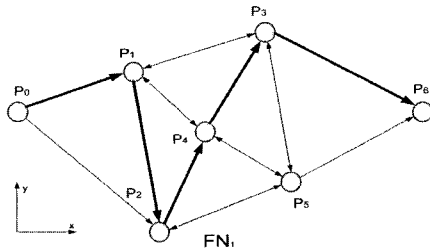


그림 11. 청소영역을 플로우네트워크로 표현.

Fig. 11. Cleaning area by flow network.

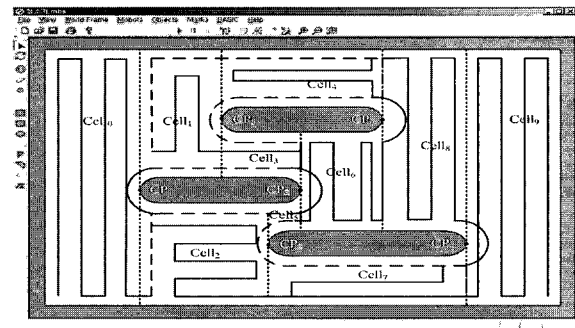


그림 12. 세개의 장애물이 있을 때 순환알고리즘을 이용한 경로계획.

Fig. 12. Path planning with three obstacles by circular algorithm.

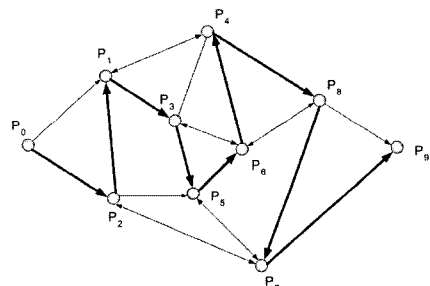


그림 13. 그림 12번 청소영역을 플로우네트워크로 표현.

Fig. 13. Flow network for fig. 12.

Table 1은 청소영역의 크기가 400×700cm 이고 로봇의 최고속도가 30cm/s일 때 기존의 방식과 제안된 방식의 전체 회전수와 시간을 나타내었다. 전체 회전 수는 제안된 방식이 기존의 방식보다 27%절약 되었고 전체 영역 청소 시 소요된 시간은 16.4%가 절약됨을 알 수 있었다. 시뮬레이션 결과 기존의 방식보다 플로우네트워크를 이용한 방식을 이용할 때 더 적은 방향전환이 발생하였고, 이로 인하여 보다 효율적인 경로계획을 함을 알 수 있었다.

그림 10은 장애물의 정보를 알 수 없는 환경에서 순환알고리즘을 이용한 전체영역경로계획을 나타내었다. 이때 순환알고리즘을 이용해 장애물의 주변에 있는 입력입계점(CP<sub>1</sub>, CP<sub>3</sub>)과 출력입계점(CP<sub>2</sub>, CP<sub>4</sub>)를 찾게 되고 입출력 입계점을 중심으로 셀 영역이 Cell<sub>1</sub>, Cell<sub>2</sub>, Cell<sub>3</sub>, Cell<sub>4</sub>, Cell<sub>5</sub>로 분할된다. 그림 10에서 순환경로는 굵은 점선으로 나타내었다. 그림 11은 플로우네트워크로 셀과 셀 사이의 경로계획을 나타낸 것으로 입력정점(P<sub>0</sub>), 출력정점(P<sub>6</sub>), 중간정점(P<sub>1</sub>~P<sub>5</sub>)으로 이루어진다. 정점간의 이동경로는 입력정점(P<sub>0</sub>)에서 x축 거리가 가장 가까운 정점으로 이동하며 한번 이동한 정점은 이동하지 않는 것을 원칙으로 한다. 정점(P<sub>1</sub>)은 정점(P<sub>2</sub>, P<sub>3</sub>, P<sub>4</sub>)로 이동할 수 있는데, 이때 입력정점(P<sub>0</sub>)과 x축 거리가 가장 가까운 정점(P<sub>2</sub>)으로 이동하게 된다. 그리고 출력정점(P<sub>6</sub>)에 도착하면 플로우네트워크의 경로계획은 완성된다. 전체경로계획의 순서를 정리하면 다음과 같다. Path={ Cell<sub>0</sub>, Cell<sub>1</sub>, Cell<sub>2</sub>, Cell<sub>4</sub>, Cell<sub>3</sub>, Cell<sub>5</sub>, Cell<sub>6</sub> }

그림 12는 장애물이 세개 있을 경우 순환알고리즘을 이용해 입출력 입계점을 찾은 후 전체영역 경로계획 함을 나타내었고, 그림 13은 그때 생성된 플로우네트워크를 나타내었다. 이동로봇이 기본 경로계획으로 이동 시 아직 입계점의 정보를 알 수 없기 때문에 Cell<sub>0</sub>영역의 기본경로가 마치는 곳에서 가까운 셀(Cell<sub>2</sub>)로 이동하게 되고 그림 13에서 입력정점(P<sub>0</sub>)에서 P<sub>2</sub>로 이동경로를 나타내었다. 순환알고리즘을 이용해 장애물(Obstacle1, Obstacle2)의 주변에 있는 입력입계점(CP<sub>1</sub>, CP<sub>3</sub>)과 출력입계점(CP<sub>2</sub>, CP<sub>4</sub>)를 찾게 되고 입출력 입계점을 중심으로 다섯 개의 셀 영역으로 분할된다. 그리고 장애물(Obstacle3)의 주변을 순환알고리즘을 이용해 입력입계점(CP<sub>3</sub>)과 출력입계점(CP<sub>6</sub>)을 추가로 찾아서 셀 영역은 다시 Cell<sub>1</sub>, Cell<sub>2</sub>, Cell<sub>3</sub>, Cell<sub>4</sub>, Cell<sub>5</sub>, Cell<sub>6</sub>, Cell<sub>7</sub>, Cell<sub>8</sub>로 8개의 셀로 분할이 이루어진다. 그림 13에서 정점간의 이동 경로계획을 보면 다음과 같다. 정점(P<sub>2</sub>)에서는 정점(P<sub>1</sub>, P<sub>3</sub>, P<sub>7</sub>)로 이동이 가능하나 입력정점(P<sub>0</sub>)과 가장 가까운 정점(P<sub>1</sub>)으로 이동하게 된다. 정점(P<sub>3</sub>)에서는 4개의 정점(P<sub>1</sub>, P<sub>4</sub>, P<sub>5</sub>, P<sub>6</sub>)으로 이동이 가능하나 입력정점(P<sub>0</sub>)과 가장 가까운 정점(P<sub>1</sub>)은 이미 방문한 정점이므로 이동하지 못하고 다음으로 가까운 정점(P<sub>3</sub>)으로 이동하게 된다. 이러한 이동조건을 충족시키면서 전체경로계획의 순서를 정리하면 다음과 같다. Path={ Cell<sub>0</sub>, Cell<sub>2</sub>, Cell<sub>1</sub>, Cell<sub>3</sub>, Cell<sub>5</sub>, Cell<sub>6</sub>, Cell<sub>4</sub>, Cell<sub>8</sub>, Cell<sub>7</sub>, Cell<sub>9</sub> }.

V. 결론

본 논문은 장애물의 정보를 알 수 없는 환경에서 청소로봇의 경로계획 시 셀 영역의 모양과 크기에 따라 셀 내부의 최소 방향전환 경로계획을 설계함으로써 기존의 방식보다 방향전환 횟수가 적은 전 영역 청소계획 방식을 제안하였다.

이동로봇이 방향전환을 하기 위해서는 우선 정지 후 회전을 해야만 됨으로 많은 시간이 소비되며 정지 후 가/감속을 하기 위해서는 많은 전력이 소비된다. 그러므로 청소할 영역의 모양에 따라 최소의 방향전환을 하도록 플로우네트웍을 이용한 셀 분할방식을 제안하였다. 넓은 청소영역을 여러 플로우네트웍으로 분리하여 해석함으로써 좀더 쉽게 경로계획을 할 수 있었다. 각 셀들 사이의 경로계획을 한 후 셀 내부의 최적의 경로계획을 함으로써 제안한 전체영역경로계획을 완성하였다. 그리고 시뮬레이션을 이용해 제안된 전체영역경로계획방식이 기존의 방식보다 효율성이 양호함을 보였다.

### 참고문헌

- [1] S. S. Ge and Y. J. Cui, "New potential function for mobile robot path planning," *IEEE Trans. Robotics and Automat.*, vol. 16, no.5, pp. 615-620, Oct. 2000.
- [2] A. Zelinsky, R. A. Jarvis and J.C. Byrne, "Planning paths of complete coverage of an unstructured environment by a mobile robot," in *Proc. IEEE Int. Conf. Robot. Automat.*, pp. 533-538, 1993.
- [3] C. Hofner and G. Schmidt, "Path planning and guidance techniques for an autonomous mobile cleaning robots," *Robotics and Autonomous Systems*, vol. 14, pp. 199-212, 1995.
- [4] G. Schmidt and C. Hofer, "An advanced planning and navigation approach for autonomous cleaning robot operations," in *Proc. IEEE Int. Conf. Intelligent Robots and Sys.*, pp. 1230-1235, 1998.
- [5] R. N. Carvalho, H. A. Vidal, P. Vieira, and M. I. Ribeiro, "Complete coverage path planning and guidance for cleaning robots," in *Proc. IEEE Int. Symp. Industrial Electronics*, pp. 677-682, 1997.
- [6] A. Zenlinsky, "A mobile robot exploration algorithm," *IEEE Trans. Robotics and Automation*, vol. 8, no. 6, pp. 707-717, 1992.
- [7] J. S. Oh, J. B. Park, and Y. H. Choi, "Complete coverage navigation of clean robot based on the triangular cell map," in *Proc. IEEE Int. Symp. Industrial Electronics*, pp. 2089-2093, 2001.
- [8] Y. Gabriely and E. Rimon, "Spanning-tree based coverage of continuous areas by a mobile robot," in *Proc. IEEE Int. Conf. Robot. Automat.*, pp. 1927-1933, 2001.
- [9] E. U. Acar, H. Choset, Y. Zhang, and M. Schervish, "Path planning for robotics demining: robust sensor-based coverage of unstructured environments and probabilistic methods," *Int. J. Robot. Res.*, vol. 22, no. 7-8, pp. 441-466, July-August, 2003.
- [10] H. Choset, "Coverage of known spaces: the boustrophedon cellular decomposition," *Autonomous Robots*, vol. 9, no. 3, pp. 247-253, 2000.
- [11] E. U. Acar and H. Choset, "Critical point sensing in unknown environments," in *Proc. IEEE Int. Conf. Robot. Automat.*, pp. 3803-3810, 2000.
- [12] E. U. Acar, H. Choset, A. A. Rizzi, P. N. Atkar and D. Hull, "Morse decompositions for coverage tasks," *Int. J. Robot. Res.*, pp. 331-334, 2002.
- [13] S. H. Nam, S. W. Suk, S. B. Moon, and K. M. Lee, "Obstacle recognition and localization for vacuum cleaning robots employing vision cameras," in *Proc. 32nd Int. Symp. Robot.*, pp. 1266-1270, 2001.
- [14] M. A. Weiss, *Data Structures and Algorithm Analysis In C*, Addison Wesley Longman, pp. 295-303, 1997.
- [15] L. Zahan and B. K. Ghosh, "Line Segment Based Map Building and Localization Using 2D Laser Rangefinder," in *Proc. IEEE Int. Conf. Robot. Automat.*, pp. 2538-2543, 2000.



남 상 현

1998년 한밭대학교 전기공학과 학사 졸업. 2000년 명지대학교 대학원 전기전자공학부 전력전자 석사졸업. 2000년~현재 세종대학교 대학원 컴퓨터공학과 박사과정 재학중. 관심분야는 청소로봇, 경로계획, Localization.



문 승 빈

1985년 명지대학교 전기공학과 학사 졸업. 1988년 University of Michigan, Ann Arbor, EECS 석사. 1993년 Purdue University, W. Lafayette, EE 박사, 1993년~1999년 삼성전자 수석연구원. 현재 세종대학교 컴퓨터 공학과 부교수. 관심분야는 로보틱스, 로봇비전, 임베디드 시스템.