

청소로봇을 위한 지능형 온라인 경로계획 알고리즘 개발

(Development of Intellectual Online Complete Coverage
Planning Algorithm for Cleaning Robot)

신 용 우 [†] 최 승 윤 ^{**} 정 태 충 ^{***}
(Yongwoo Shin) (Seungyoon Choi) (Taechoong Chung)

요 약 청소로봇은 장애물을 피하며 전체 바닥공간을 청소하여야 한다. 온라인 커버리지 경로계획이란 벽과 장애물의 위치를 미리 알지 못하는 상황에서 청소를 진행하며 지도를 그려가면서 경로계획 및 청소를 하는 방식을 말한다. 본 논문에서는, SIBA(Spiral Intellectual Boustrophedon and A* search) 경로계획 알고리즘을 제안하였다. 기존의 알고리즘은 나선형 알고리즘으로 벽과 장애물의 위치를 알아내며, 나머지 바닥부분은 B-알고리즘(Boustrophedon algorithm)으로 채워나가는 방식이었다. 그런데 B-알고리즘은 영역을 여러 조각을 내는 문제와 청소하는 영역이 차례로 진행되지 않는다는 단점이 있었다. 본 연구에서는 가급적 영역의 개수를 줄이면서도 한쪽부터 차례대로 청소를 진행하도록 B-알고리즘을 사람이 하는 방식으로 고려한 IB(Intellectual Boustrophedon) 알고리즘으로 개선하였다. 이 방식은 매우 새로운 개념이다. 따라서 SIBA알고리즘은 다른 알고리즘에 비해서 청소영역의 개수를 줄이고, 전체적으로 좌에서 우로 순차적으로 청소할 수 있게 하여, 청소로봇의 행위가 실제 사람의 청소방식과 유사하게 하였다.

키워드: 청소로봇, 경로계획, 알고리즘, 로봇, 온라인

Abstract Cleaning robot has to sweep whole floor under the given walls and obstacles. Online coverage path planning is the way to sweep without any information about walls and obstacles beforehand. This means that the robot creates a map while sweeping. In this paper, we propose SIBA(Spiral Intellectual Boustrophedon and A* search) online coverage path planning algorithm. The existing algorithm use spiral algorithm to find walls and obstacles and it use Boustrophedon algorithm to fill in the rest regions. But Boustrophedon algorithm makes many fractions. Moreover, sweeping regions are somewhat random. In order to improve their drawbacks, we propose Intellectual Boustrophedon algorithm which mimics human behavior by tuning Boustrophedon algorithm. This algorithm reduces the number of cleaning regions and proceeds to clean from left to right sequentially. This idea is very new. SIBA algorithm shows better performances from the 2 viewpoints compared to other algorithm: The number of sweeping regions and better moving style (moving from left to right region) like human.

Keywords: cleaning robot, coverage planning, algorithm, robot, online

· This research was also partly supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science, and Technology (2010-0012609).

[†] 정 회 원 : 동아방송예술대학교 디지털영상과 교수
littlelspmk@hanmail.net

^{**} 정 회 원 : 경희대학교 컴퓨터공학과
sychoi84@khu.ac.kr

^{***} 종신회원 : 경희대학교 컴퓨터공학과 교수
tcchung@khu.ac.kr
(Corresponding author)

논문접수 : 2013년 5월 21일

심사완료 : 2013년 7월 24일

Copyright©2013 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지 : 소프트웨어 및 응용 제40권 제10호(2013.10)

1. 서론

로봇기술이 발달함에 따라 인간의 생활에 보다 가까워지고 다양한 분야에서 로봇 연구가 진행되고 있다. 로봇은 산업용, 무인탐사 또는 노인이나 장애가 있는 사람들의 보조역할 등의 다양한 서비스 분야에서 인간과 상호작용하며 다양한 형태로 이용된다[1].

일상생활에서 사용되고 있는 대부분의 로봇들은 주어진 환경 내에서 이동하며 인간을 보조하는 서비스용 이동 로봇이다. 일반적인 이동로봇이 원하는 위치로 이동하는 가장 빠른 경로를 찾는다면, 청소로봇은 주어진 환경을 모두 청소하고 지날 수 있는 가장 적합한 경로를 찾는다. 이처럼 모든 영역을 지나는 경로를 찾는 것을 커버리지 경로계획(Coverage Path Planning)이라 한다.

가정용 청소 로봇이 개발된 후, 가정에서 가장 많이 사용되는 가정용 서비스 로봇이 되었다. 하지만 청소 로봇의 성능은 사용자들에게 만족감을 주지 못하고 있는데, 그 이유 중 하나가 청소 주행 성능이 기대에 못 미치기 때문이다. 전 영역 청소를 완전하게 하기 위해서는 로봇의 정확한 위치 정보와 청소 환경에 대한 지도를 생성하거나 이용할 수 있어야 한다. 물론, 레이저 센서와 같이 먼 거리를 정확하게 측정할 수 있고, 이것으로 얻어진 정보들을 빠르게 처리할 수 있는 처리 장치가 있다면 해결할 수 많은 방법들이 존재한다. 하지만 일반적인 가정용 청소 로봇에는 비용 문제와 로봇의 구조적인 문제로 인해 사용할 수 있는 센서들이 제한적이고, 그 성능 또한 좋고 할 수 없다. 또한 기존에 많이 알려져 있는 위치 인식과 지도 생성 방법들은 청소 로봇을 위해 사용되는 연산 장치로는 감당 할 수 없을 정도로 복잡하고 많은 계산량을 요구한다. 그러므로 일반적인 가정용 청소 로봇을 위해, 사용하는 최소한의 거리 측정 센서와 연산 장치를 고려해서, 실제 적용 가능 할 뿐 아니라, 청소 성능을 최대화 할 수 있는 알고리즘이 필요하다[1].

기존의 커버리지 경로계획 알고리즘들은 구현의 간편함 때문에 무작위로 이동을 하는 랜덤 알고리즘을 사용했다. 그러나 영역을 모두 커버 할 수 없거나, 많은 시간을 필요로 했다. 주어진 환경을 모두 커버 할 수 없는 랜덤 기반 알고리즘의 문제를 해결하기 위해 완벽한 커버를 목적으로 하는 많은 연구가 진행되었다[2,3].

진행된 커버리지 연구들은 로봇이 환경에 대한 정보를 가지고 있고 시작위치와 종료위치를 알고 있을 때 모든 영역을 커버할 수 있는 이동경로를 생성할 수 있다[4].

그러나 로봇이 사전에 환경에 대한 정보를 가지고 있지 않을 때는 기존연구의 알고리즘을 적용할 수 없는 문제점을 가지고 있다. 이런 문제점 때문에 로봇이 사전 정보를 가지고 있지 않은 상태에서 영역을 커버하기 위

해서는 로봇 스스로 환경에 대한 정보를 수집하는 연구가 진행되었다. 환경에 대한 지도를 로봇이 스스로 생성한다면, 생성된 지도를 기반으로 커버리지 알고리즘을 적용하여 모든 영역을 청소할 수 있다.

본 논문에서는 환경에 대한 사전정보를 가지고 있지 않은 상태에서 이동로봇이 대상공간을 커버할 수 있는 SIBA(Spiral, Intellectual Boustrophedon and A* search) 커버리지 경로계획 알고리즘을 제안한다. 알고리즘의 효율성을 확인하기 위해 기존에 출시된 온라인 알고리즘들과 비교하였다. 청소영역의 개수를 줄이고, 실행시간을 줄이며, 불필요한 이동을 줄여 사람이 청소하는 형태로 청소가 이루어짐을 알 수 있었다. 성능 면에서 향상된 점을 알 수 있었다.

본 논문의 구성은 1장 서론에 이어 2장에서는 관련연구에 대해 살펴보고 3장에서는 본 논문에서 제안하는 SIBA 알고리즘에 대해 설명한다. 4장에서는 실험 및 결과에 대해 알아본다. 마지막으로 5장에서는 최종 결론을 맺도록 한다.

2. 관련 연구

임의 경로 진행 방법(Heuristic and Random Approaches)에서는 로봇이 경로를 계획하지 않는다. 단순히 방향을 임의로 설정하고 미리 정해진 규칙에 따라 진행 방향을 재설정한다[5]. 이 방법들 중 하나로, 로봇이 미리 규정된 간단한 행동 세트(Set of behaviors)를 갖도록 한 방법이 있다. 행동세트를 가지고 있는 로봇은 주어진 영역에서 로봇이 진행 가능한 자유 공간을 진행하다가 장애물을 만나거나 벽에 부딪히는 등의 특정기준을 만나게 되면 행동세트에서 알맞은 동작을 선택하고 이를 수행한다. 실제로 이런 방법을 사용해서 어느 정도 안정된 결과를 얻을 수 있다. 그러나 이 방법으로는 커버리지 임무를 완료할 수 없다는 단점이 있다.

오프라인(Off-line) 경로계획방법은 주어진 지도를 분석해서 진행할 커버리지 경로를 미리 생성하는 방법이다. 이 방법은 주어진 지도를 바탕으로 하기 때문에 커버리지 임무를 완벽히 수행할 수 있는 경로를 생성할 수 있다. 이 방법에서는 먼저 영역을 구분하고 전체 커버순서를 정한 다음 각각의 작은 영역의 경로를 생성하는 과정을 통해 커버리지 경로를 생성한다. 오프라인 경로계획으로 최적화된 커버리지가 가능하나 지도를 분석해서 커버리지 경로를 사전에 생성해야 하는 단점이 있다.

오프라인 경로생성방법의 접근방법으로는 셀(Cell) 분할을 이용한 경로계획 방법이 있다. 이 방법에서는 영역 구분 작업을 통해 주어진 영역을 여러 개의 작은 셀들로 나눈 다음 각 셀 영역을 이동로봇의 간단한 동작으로 커버한다[6]. 셀 분할 방식은 장애물의 변화에 따라

셀과 셀 사이의 경로계획을 함으로써 전체 영역을 커버리지 할 수 있다. 주어진 공간의 장애물을 고려하여 이동이 불가능한 영역을 표시하고 전체 영역을 이동 가능한 셀과 이동이 불가능한 셀로 분할한다. 이는 복잡한 문제를 여러 개의 간단한 문제로 나누어서 해결하는 방법이다. 실제 환경을 위해서는 주어진 환경의 복잡성을 고려할 필요가 있다. 실제 환경의 경우 벽 또는 가구로 인한 선 성분이 많으면서도 잡음이 산재하는 환경일 가능성이 크다. 이를 고려해 안정된 영역구분을 수행하면서도 복잡성을 감소시킬 수 있는 영역구분 방법이 필요하다. 셀 분할 방식은 장애물의 변화에 따라 청소영역을 큰 셀로 나누어서 해석하는 경로계획을 말한다. 전체 청소영역은 큰 셀로 표현되고 셀과 셀 사이의 경로계획을 함으로써 계획된 모션을 줄일 수 있다[7,8].

또 다른 방법으로는 모스 함수(Morse function)를 이용한 방법이 있다. 이 방법은 임계점(Critical point) 사이의 기하학적 특성을 통하여 완벽한 커버리지를 보장한다. 이렇게 나누고 정복하는 방법은 작게 나누어진 각각의 셀을 커버하는 것을 통해 전체 영역에 대한 완벽한 커버리지를 보장할 수 있는 매우 유용한 방법이다[4].

시간적으로 최적화된 커버리지 경로에 초점을 맞춘 영역구분 방법이 제안됐다. 이 방법에서는 이동로봇의 방향 전환횟수가 최적화의 기준으로 사용된다. 이동로봇의 특성상 로봇이 방향을 전환할 때 많은 시간이 소요되는 것을 고려한 방법이다. 이 방법은 최적의 경로를 생성할 수 있지만, 모든 가능성에 대해 경로를 생성해보는 방법이기 때문에 간단한 환경에 대해서도 너무 많은 계산시간이 필요하다는 단점이 있다[4].

온라인(On-line) 경로 계획방법은 미리 지도를 알지 못하는 상태에서 센서 정보를 분석해서 커버리지 경로를 생성하는 방법이다. 이를 위한 방법으로 지도의 기하학적 구조를 바탕으로 영역을 구분하고 이를 분석해서 커버리지를 수행하는 방법이 있다. 영역구분 결과는 기하학적 지도로 표현되고 로봇은 각각의 셀을 커버한다.

온라인 경로계획 방법으로는 임의경로진행방법이 있다[5]. B-동작과 A* 알고리즘을 사용하여 온라인으로 청소 대상이 되는 영역의 경로를 생성한 방법이 있다[9]. 또 다른 방법으로는 B-동작, S-동작, A* 알고리즘을 적절히 혼용하여 사용하는 HSB 알고리즘을 제안한 논문이 있다[10].

3. SIBA 알고리즘

SIBA(Spiral, Intellectual Boustrophedon and A* search) 알고리즘은 S-동작(Spiral motion)과 개선된 B-동작이 혼합된 형태이다. 그림 1은 표 1의 실험환경 1의 결과이다.

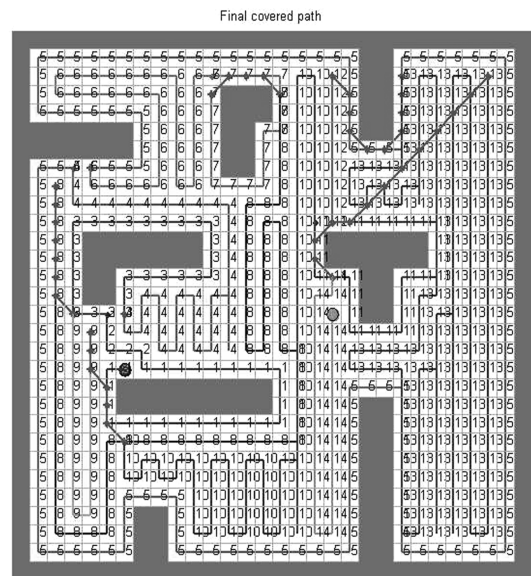


그림 1 최종적으로 청소가 완료된 SIBA 알고리즘

Fig. 1 Completely cleaned by the SIBA algorithm

S-동작은 시작지점에서부터 외곽을 경유하여 중앙으로 진입한다. 그림 1에서 보는 바와 같이 전체영역의 외곽지역의 경계부분(Boundary)은 S-동작으로 처리하였다. 5의 순번으로 처리된 부분이다. 장애물 역시 S-동작으로 처리하였다. 상황에 따라 시계방향 또는 시계반대 방향으로 돌면서 처리한다. 1, 3, 7, 11번으로 번호가 배정되었고 이 번호는 처리된 순서를 의미한다.

나머지 부분은 IB-동작(Intellectual Boustrophedon motion)에 의해 처리되었다. 더 이상 청소할 영역을 찾을 수 없을 때는 A* 길 찾기 알고리즘을 이용하여 미청소영역으로 안내한다. 빈곳 없이 청소가 완료되었다.

로봇은 기본적으로 상하로 이동하면서 청소를 한다. 로봇의 위치는 이동 중 장애물을 만나거나 벽에 부딪혔을 때에 방향을 전환하게 된다. 또한 한번 청소한 곳에 다다르게 되면 청소하지 않은 곳으로 이동하여 청소를 계속한다. 위치의 오차가 발생하여도 현재위치에서부터 다시 이동하면서 청소를 한다. 오차가 발생하여 청소를 하지 못한 영역은 나중에 다시 이동하여 청소를 하게 된다 그러므로 알고리즘의 수행이나 전체영역을 커버하는 데에는 문제가 없다.

3.1 작업영역 찾기에 사용될 A* 알고리즘

A* 알고리즘은 두 지점 사이의 최단거리를 가진 경로를 효율적으로 찾는데 효과적인 알고리즘이다. 이 알고리즘은 지금까지 온 거리와 남아있는 거리의 저평가 값을 합친 값을 기준으로 다음에 확장할 노드를 선택하는 방식이다[9].

표 1 SIBA 알고리즘과 HSB, BA* 알고리즘의 성능비교

Table 1 The performance comparison of the SIBA, HSB and BA* algorithm

Experimental environment	SIBA			HSB			BA*		
	B. motion	Out of sequence	Path length	B. motion	Out of sequence	Path length	B. motion	Out of sequence	Path length
1	9	2	770	17	6	804	14	4	781
2	7	0	706	10	5	739	8	4	716
3	6	0	796	13	3	804	11	2	799
4	9	1	820	16	7	856	12	8	822
5	10	3	913	17	7	938	11	5	920
평균	7.8	1	793	14.4	5.4	820.2	10.8	4.4	798.6

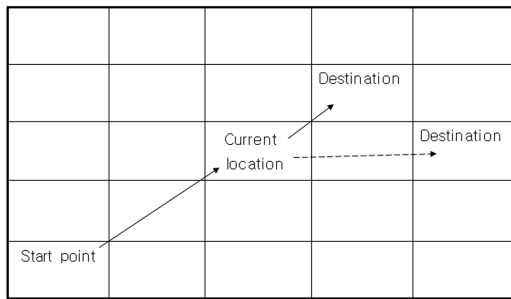


그림 2 A* 길 찾기 알고리즘의 상황표현

Fig. 2 A* search algorithm

이 알고리즘을 청소로봇에 응용을 할 때는 다음과 같이 적용한다.

모든 셀 중에서 인접된 4면의 셀 중에서 1-3개 면의 셀이 청소되지 않은 셀들은 백트래킹 리스트(Backtracking list)에 들어간다. S-동작이나 B-동작을 할 때 그런 조건을 만족하는 옆 셀을 백트래킹 리스트에 넣는다. 한편 청소가 된 셀들은 백트래킹 리스트에서 제외된다.

청소를 할 때는 4개 방향으로만 움직이지만 백트래킹 할 때는 대각선 방향으로도 움직일 수 있도록 한다. 휴리스틱 함수는 실제 거리보다 저평가되어야하므로 두 지점 사이의 직선거리를 휴리스틱함수로 사용한다.

청소로봇이 인접 셀들이 모두 청소되었거나 장애물인 상황에 도달하면 A* 탐색을 시작한다.

A* 알고리즘은 일반적으로 목표지점이 하나인 경우에 사용한다. 그러나 본 논문의 경우 그림 2와 같이 목표지점이 백트래킹 리스트의 모든 셀들이 대상이다. 그러므로 시작점에서 여러 곳의 목표지점의 셀 중 가장 가까운 한 곳으로 이동하여야 한다. 따라서 남은거리 계산할 때는 백트래킹 리스트에 있는 모든 셀들 중 가장 가까이에 있는 셀을 기준으로 계산하면 된다.

3.2 자료 구조

이 알고리즘은 다음의 자료구조를 관리한다.

벽이나 장애물을 표시하고, 청소를 한 부분인지 아닌

지를 알려주는 청소상황 매트릭스이다. 여기서 한 가지 고려할 점은 S-동작으로 청소된 부분은 구분해서 표시한다. S-동작으로 청소된 부분은 영역의 개수를 줄이기 위해 IB-동작으로 청소를 한 번 더 해도 될 수 있도록 하기 위함이다.

모든 셀 중에서 인접된 4면의 셀 중에서 1-3 개 면의 셀이 청소되지 않은 셀들은 백트래킹 리스트에 들어간다. S-동작이나 B-동작을 할 때 그런 조건을 만족하는 옆 셀을 백트래킹 리스트에 넣는다. 한편 청소가 된 셀들은 백트래킹 리스트에서 제외된다.

그림 3에 대한 설명은 다음과 같다.

0 단계: 자료구조를 초기화한다. 바닥 지도 매트릭스 및 백트래킹 리스트를 비운다. 온라인 청소 알고리즘이므로 벽과 장애물의 위치를 모르는 상태에서 시작하여 S-동작을 통해 벽과 장애물의 위치를 알아낸다. 따라서 S-동작 전까지는 벽과 장애물의 위치를 모르므로 초기엔 비어있게 된다. 지도 매트릭스에 각 셀에 들어갈 수 있는 값은 벽이나 장애물, S-동작청소, IB동작 청소, 청소가 안된 상황 등을 표시할 수 있는 값들이다.

1단계: 로봇이 처음에 벽에 붙어서 놓여 있다고 가정한다. 청소기는 벽을 따라서 S-동작을 통해 전체영역의 외곽지역 경계부분을 찾아내며 청소를한다. 이때 진행하며 자료구조를 수정해 나간다.

2단계: 인간이 청소하는 형태의 IB-동작을 진행하며 자료구조를 수정해 나간다. 진행해 나가다가 장애물을 만나거나 막다른 골목에 도착하는지 검사한다. 만일 장애물이 나타나면 그곳에서부터 S-동작으로 장애물의 경계선을 찾아가며 테두리를 청소한다. 장애물의 경계선을 다 청소하면 그 자리가 임계점이 되어 다음 IB-동작 영역을 찾도록 한다.

여기서 장애물을 시계방향 혹은 반시계방향으로 이동할지는 상황에 따라 방향을 결정한다. 로봇의 좌측 혹은 우측에 장애물이 접하면 그 방향으로 따라가며 돌면 된다. 그렇지 않고 로봇의 전면에 장애물이 만나면 진행방향의 오른쪽으로 도는 것으로 한다.

```

Step 0: Initialize Data structure
        (Layout Map matrix and Backtracking list)

Step 1: Robot sweeps boundary of workspace by S-motion with updating data structure
Repeat
  Step 2: It perform a Intellectual Boustrophedon motion with updating data structure until
          reach to a real obstacle or gets stuck in a critical point or dead end
    Step 2.1: If the robot reaches to an obstacle:
      - Sweep obstacle boundary with updating data structure
      - Set the end point of S-motion critical point

Step 3: A* algorithm finds the path to the nearest backtracking point
        from critical point or dead end.
Step 4: Move to the backtracking point

```

그림 3 SIBA 알고리즘

Fig. 3 SIBA algorithm

3단계: 막다른 골목을 만났거나 S-동작을 하고 난 후 그 자리에서 다음 IB-동작을 할 영역 시작점을 찾아야 한다. A* 알고리즘을 이용해 가장 가까운 백트래킹 지점과 그곳으로 가는 경로를 찾는다.

4단계: 찾은 경로를 따라 로봇을 이동하여 새로운 IB-동작을 한다.

3.3 IB-동작

B-동작은 상하 운동만 하며 상하 운동 중에 벽이나 장애물 혹은 이미 청소한 곳을 만나면 일단 90도 회전하여 한 칸만 진행한 후 북쪽이 비어있으면 북쪽으로

진행하고 북쪽이 막혀있으면 남쪽으로 진행하는 방식이다. 그런 방식은 작은 영역을 많이 만들어 내고 왼쪽부터 차례대로 청소하는 목표와는 관계없이 진행하게 된다. 따라서 B-동작을 다음과 같이 개선하여 작은 작업 영역의 개수를 줄이고 왼쪽부터 차례로 진행되도록 IB-동작을 설계해 보자.

그림 4의 북쪽과 남쪽방향은 상하동작을 의미한다. 서쪽이란 좌측방향을 의미하고 동쪽은 우측방향을 의미한다. 동서남북을 의미하는 west, south 등은 로봇 좌표계이다.

상하로만 움직이면서 장애물을 만나면 반대편으로 이동하는 점을 개선하였다. 상하로만 움직이는 경우에는 좌측 모서리부분에 미 청소영역이 발생하게 되어있다. 이렇게 작은 영역들이 많이 발생하면 백트래킹을 이용하여 나중에 처리하게 된다. 그래서 다음과 같이 처리하였다. B-동작 중 좌측이 비어있다면 좌측으로 수평 진행한다. 예를 들어 그림 1의 6번 동작의 경우 좌측 상단부를 보면 좌측 공간이 비어있다. 그래서 좌측공간을 먼저 처리하고 우측으로 IB-동작을 진행한다.

또한 B-동작의 특성상 북쪽으로 이동하여야 하지만 남쪽이 비어있어 작은 영역의 발생이 우려될 때는 남쪽 영역을 중복 청소하고 북쪽으로 진행한다. 그림 1의 4번, 10번 동작에서 발생하였다.

상하로 진행하는 로봇의 앞에 S-동작으로 청소된 셀이 있다면 그 옆 열이 비어있는 경우는 옆 열로 가지 않고 이미 S-동작으로 청소가 되었어도 그곳을 지나간다. 옆 열로 진행하면 옆 열의 남쪽에 작은 영역이 추후에 생기게 되므로 S-동작 부분을 IB-동작으로 다시 지나가게 한다.

SIBA 알고리즘에서는 B-동작의 단점을 개선하였다. B-동작을 수행하는 중에 좌측에 빈 영역을 발견하면 청소하고 나오는 방식으로 처리한다. 그래서 전체적인 영

```

if (south and west directions are not blocked)
  move west;
else
  if (north and west directions are not blocked)
    move west;
  else
    if (north and south directions are not blocked)
      move south;
    else
      if (south direction in the spiral area is not
        blocked)
        move south;
      else
        if (north direction is not blocked)
          move north;
        else
          if (south direction is not blocked)
            move south;
          else
            if (east direction is not blocked)
              move east;
            else
              if (west direction is not blocked)
                move west;

```

그림 4 IB 알고리즘

Fig. 4 Intellectual Boustrophedon algorithm

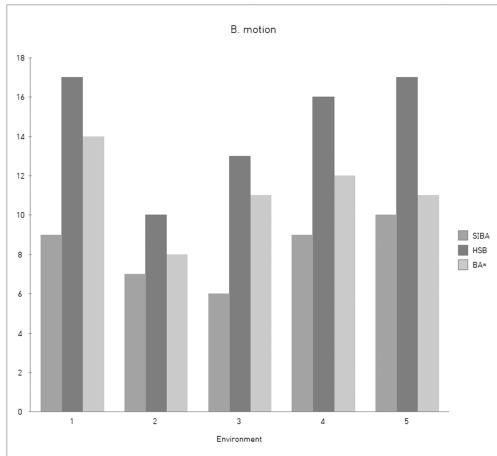


그림 5 SIBA 알고리즘과 HSB, BA* 알고리즘의 B. motion 비교

Fig. 5 The B. motion comparison of the SIBA, HSB and BA* algorithm

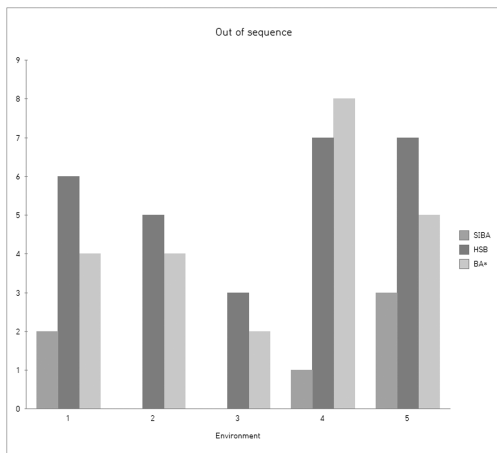


그림 6 SIBA 알고리즘과 HSB, BA* 알고리즘의 순서이탈 비교

Fig. 6 The Sequence comparison of the SIBA, HSB and BA* algorithm

역의 수를 줄이고 순차적인 처리가 되도록 한다. 그리고 북쪽으로 이동하여야 하지만 남쪽에 청소할 곳이 있을 때에는 남쪽영역을 먼저 처리함으로써 작은 빈 영역이 발생하는 것을 사전에 방지할 수 있다.

4. 실험 및 결과

그림 1을 보면 영역의 수가 많지 않은 것을 알 수 있다. 영역이 14개인 것을 알 수 있다. 이것은 B-동작의 단순성으로 많은 조각이 날 수 있던 영역을 연결 동작

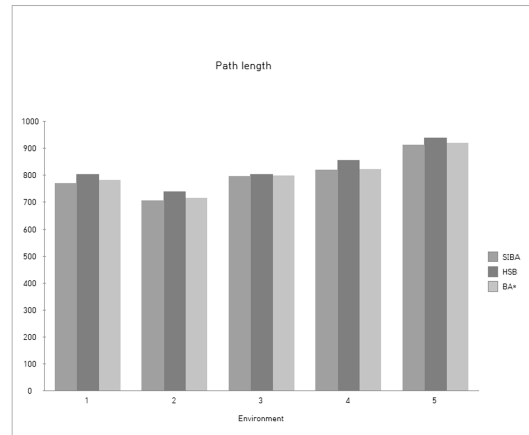


그림 7 SIBA 알고리즘과 HSB, BA* 알고리즘의 경로 길이 비교

Fig. 7 The Path length comparison of the SIBA, HSB and BA* algorithm

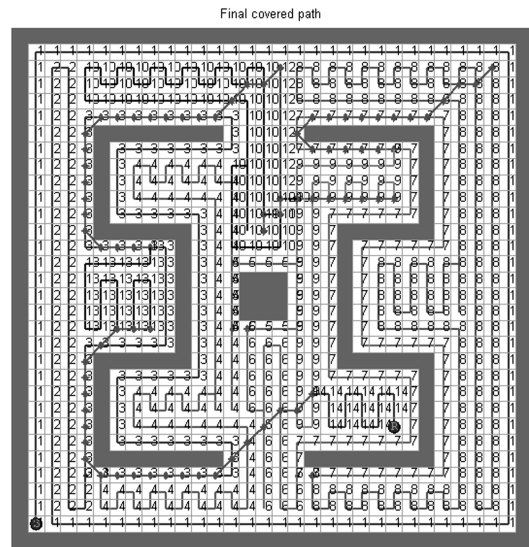


그림 8 SIBA 알고리즘의 실험결과

Fig. 8 The experimental result of the SIBA algorithm

으로 잘 처리하였기 때문이다. 또한 영역과 영역을 이어주는 백트래킹도 상대적으로 많이 발생하지 않았다.

본 알고리즘과 비교를 위해 최신의 유사한 알고리즘인 BA*와 HSB를 간단히 소개한다[9,10]. 그림 10처럼 BA*는 순수 B동작만으로 처리하는 방식이고, 그림 9처럼 HSB는 S동작과 순수B동작을 결합해 처리하는 방식이다. 그림 8은 같은 환경에 대해 SIBA 알고리즘을 사용한 결과인데, SIBA는 S동작과 IB동작의 결합을 통한 처리 방식으로 개선된 것이다.

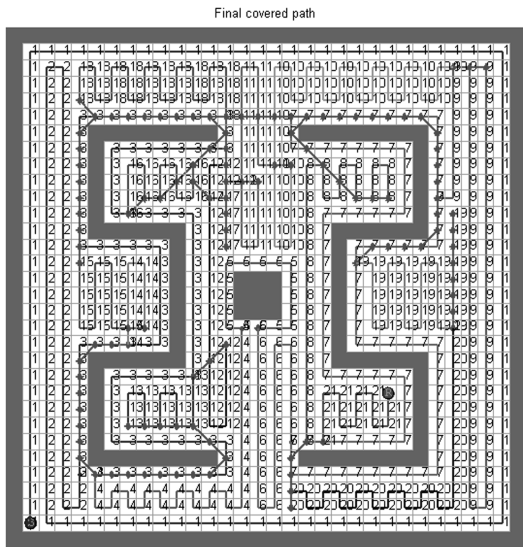


그림 9 HSB 알고리즘의 실험결과

Fig. 9 The experimental result of the HSB algorithm

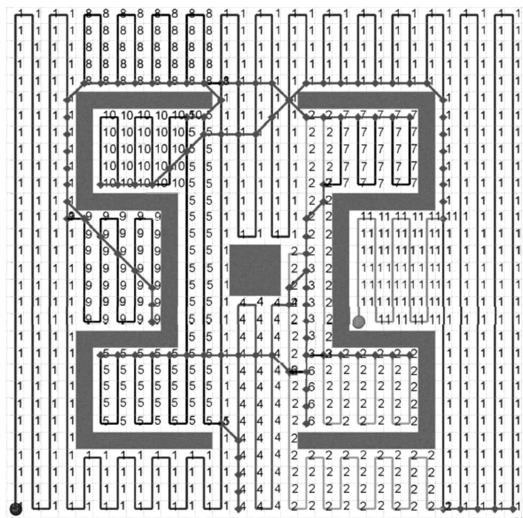


그림 10 BA* 알고리즘의 실험결과

Fig. 10 The experimental result of the BA* algorithm

HSB, BA* 알고리즘과 SIBA 알고리즘의 성능 비교 결과는 표 1과 같다. 실험환경은 5개이다. B-동작의 수는 영역의 수를 의미한다. 전체 작업영역에서 영역의 수가 적다는 것은 큰 영역들로 구성되어 있다는 것이다. 작은 영역들로 구성되었다면 백트래킹도 많이 발생하고 전체영역을 청소하는 데의 시간도 많이 소모하게 된다. 또한 큰 영역들로 구성되어 있어야 사람이 청소하는 것처럼 순차적인 청소가 가능해진다. HSB 알고리즘

의 경우 B-동작의 수의 평균은 14.4이고 BA* 알고리즘은 10.8이지만 SIBA 알고리즘은 7.8로서 작은 영역을 많이 없앴 것을 알 수 있다. 순서이탈(out of sequence)이란 좌측에서 우측으로 순차적으로 청소해야 하는 원리를 어긴 것이다. 이것은 사람이 한쪽 방향에서부터 순차적으로 청소를 하는 원리를 측정하는 지표로서 중요하다. 순서이탈 기준은 기존에 다른 논문에서 다루지 않은 본 논문이 세계적으로 처음 제안하는 실험기준이다. 본 실험에서는 좌측에서부터 번호순서대로 청소한다. 그러므로 좌측부터 번호가 순차적으로 배열되어야 한다. 순서이탈이 1이라는 것은 왼쪽의 숫자가 오른쪽보다 큰 것이 한번 있다는 뜻이다. SIBA 알고리즘에서는 순서이탈이 1이지만, HSB 알고리즘에서는 5.4이고 BA* 알고리즘에서는 4.4으로 많은 차이를 보이는 것을 알 수 있다. 또한 경로길이에서도 793을 기록하고 있다. HSB 알고리즘에서는 820.2이고 BA* 알고리즘에서는 798.6으로 차이를 보이는 것을 알 수 있다. 그러나 좌측에서 우측으로 청소를 하는 경우에 좌측 끝에서 청소를 시작한 경우에는 문제가 없지만, 그림 1과 같이 작업영역 중간에서 시작하는 경우에는 좌측 끝으로 이동하는 불필요한 이동이 발생하는 경우가 있을 수 있다.

그림 8, 그림 9, 그림 10에서는 세 가지 각 알고리즘의 실험결과를 볼 수 있다. 표 1에서의 실험환경 5의 실험결과이다.

5. 결론

청소로봇은 주어진 환경을 모두 청소하고 지날 수 있는 가장 적합한 경로를 찾을 필요가 있다. 기존의 커버리지 경로계획 알고리즘들은 구현의 간편함 때문에 무작위로 이동을 하는 랜덤 알고리즘을 사용했다. 그러나 영역을 모두 커버할 수 없거나, 많은 시간을 필요로 했다. 주어진 환경을 모두 커버 할 수 없는 랜덤 기반 알고리즘의 문제를 해결하기 위해 완벽한 커버를 목적으로 셀 분할 알고리즘이 발표되었다. 그러나 전체영역을 청소하기는 하지만, 사전에 주어진 지도를 분석하여 경로를 미리 생성하여야 했다. 그래서 사전 경로생성 단계가 없이 온라인으로 직접 경로를 생성하고 사람이 청소하는 것처럼 주변의 빈곳을 우선 처리하는 알고리즘이 필요하게 되었다.

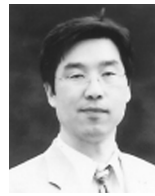
본 논문에서는 환경에 대한 사전정보를 가지고 있지 않은 상태에서 SIBA 알고리즘을 이용하여 청소로봇이 전체영역을 청소하는 과정에서 비효율적인 면을 없애고자 하였다. 기존에 발표된 알고리즘이 전체영역을 청소하는 것에 초점이 맞추어 졌다면 제안하는 알고리즘에서는 B-동작을 개선하여 인간이 청소하는 것처럼 IB-동작을 채용하여 다음과 같이 효율적으로 청소를 하였다.

첫째, IB-동작을 통해 작업영역의 개수를 줄였다. 둘째, 그로인해, A* 알고리즘의 사용빈도 수를 줄이므로 경로길이를 단축시켰다. 셋째, B-동작이 빈 영역을 발생하는 점을 개선하여 IB-동작은 왼쪽에 빈 영역이 있으면 그곳을 먼저 청소하고 나오는 방식으로 처리해 왼쪽부터 차례대로 청소하는 효과를 본다.

알고리즘의 효율성을 확인하기 위해 기존에 출시된 HSB, BA* 알고리즘과 비교 실험하였다. 성능 비교 결과 B-동작의 수, 좌에서 우로 청소하는 순서이탈, 경로 길이 면에서 전체적으로 우수한 성능을 내는 것을 알 수 있었다. 순서이탈 실험항목은 본 논문에서 세계적으로 처음 제안하는 실험항목으로 중요하다.

References

- [1] Sanghoon Baek, Tae-Kyeong Lee, Se-Young Oh and Kwangro Ju, "Feature Map Based Complete Coverage Algorithm for a Robotic Vacuum Cleaner," *Journal of Korea intelligent system society*, vol.20, no.1, pp.81-87, 2010.
- [2] Choset, H., "Coverage for robotics - a survey of recent results," *Annals of Mathematics and Artificial Intelligence*, vol.31, no.1-4, pp.113-126, 2001.
- [3] Choset, H., "Coverage of known spaces: The boustrophedon cellular decomposition. autonomous," *Robots*, vol.9, no.3, pp.247-253, 2000.
- [4] E.U.Acar, H. Choset, A.A. Rizzi, P.N. Atkar, D. Hull, "Morse decompositions for coverage tasks," *The International Journal of Robotics Research*, vol.21, no.4, pp.331-344, 2002.
- [5] R.A.Brooks, "A robust layered control system for a mobile robot," *IEEE Journal of Robotics and Automation*, vol.ra-2 no.1, mar. 1986.
- [6] W. H. Huang, "Optimal line-sweep-based decompositions for coverage algorithms," in *Proc. of the IEEE International Conference on Robotics and Automation*, Seoul, Korea, pp.27-32, 2001.
- [7] S. Koenig, M. Likhachev, "Fast replanning for navigation in unknown terrain," *IEEE Trans. on Robotics*, vol.21, no.3, 2005.
- [8] H. M. Choset et al, principles of Robot Motion, MIT press Cambridge, 2005.
- [9] Ntawumenyikizaba Abdallah, Hoang Huu Viet, TaeChoong Chung, "An online complete coverage algorithm for cleaning robots based on boustrophedon motions and A* search," *2012 8th International Conference on Information Science and Digital Content Technology*, vol.2, pp.401-405, 2012.
- [10] Seung Yoon Choi, "The Coverage path planning algorithm for cleaning robot considering obstacles in unknown environments," *M.S thesis, Dept, of Computer science*, kyung hee university, 2012.



영상

신 용 우

2004년 경희대학교 컴퓨터공학과 지능시스템전공(박사수료). 1990년~2000년 프리랜서 게임프로그래머, LG U+. 2010년~2011년 디마웍스 대표. 2000년~현재 동아방송예술대학교 디지털영상과 교수 관심분야는 게임제작, 인공지능, 디지털



최 승 윤

2012년 경희대학교 컴퓨터공학과 석사 2012년~현재 경희대학교 컴퓨터공학과 박사과정. 관심분야는 기계학습, 모바일 로봇 경로계획



정 태 중

1980년 서울대학교 전자공학과(학사). 1982년 한국과학기술원 전산학과(공학석사) 1987년 한국과학기술원 전산학과(공학박사). 1987년~1988년 KIST 시스템공학센터 선임연구원. 1988년~현재 경희대학교 컴퓨터공학과 교수. 관심분야는 기계학습, 최적화, 인공지능, 지능형로봇