

셋 커버 알고리즘을 이용한 효율적인 로봇 청소 알고리즘

전 흥 석*

Efficient Robot Cleaning Algorithm based on Set Cover Algorithm

Heung Seok Jeon *

요 약

본 논문에서는 가장 덜 복잡한 공간부터 청소하는 공간 효율적인 새로운 로봇 청소 알고리즘인 SetClean 알고리즘을 제안한다. 청소 완료 시간이 길어지거나, 예측하기 어려운 경우에는 전체 청소 완료 시간을 최적화하기 보다는 가능한 빠른 시간에 가장 넓은 공간을 최대한 청소하는 것이 유리한 경우가 있다. 이를 위해 SetClean 알고리즘에서는 전체 공간을 셋 커버 알고리즘을 이용하여 청소 가능한 공간으로 구분하고, 단위 시간 당 청소 효율이 가장 높은 공간부터 청소를 진행하게 된다. SetClean 알고리즘은 해당 청소 가능 구역의 면적뿐만 아니라 로봇의 현재 위치로부터 해당 청소 구역까지의 이동 거리, 청소 구역 내에서의 로봇의 회전으로 인한 지연 시간 등을 고려하여 최적의 청소 순서를 결정한다. 실험을 통해 SetClean 알고리즘의 동작 과정 및 성능을 보여준다.

Abstract

In this paper, we propose a new robot cleaning algorithm, which we call SetClean. The new algorithm cleans from the most less complex area. Sometimes, when the cleaning completion time can be longer or can not be estimated, cleaning larger area first is better than optimizing the whole time for cleaning. To do this, SetClean algorithm divides the whole area into cleanable sub-areas using Set Cover algorithm and cleans the area in the order of high efficiency that maximize the cleanable area per unit time. SetClean algorithm decides the navigation flow by considering not only the size of the area but also the distance from the current robot location to the area to be cleaned and the delay time caused by the number of turns within the area. The experimental results show the mechanism and performance of the SetClean algorithm.

▶ Keyword : 셋 클린(SetClean), 지능형 로봇(Intelligent Robot), 청소 알고리즘(Cleaning Algorithm)

• 제1저자 : 전흥석

• 접수일 : 2008. 3. 23, 심사일 : 2008. 4. 18, 심사완료일 : 2008. 5. 24.

* 건국대학교 컴퓨터소프트웨어전공 부교수

※ 이 논문은 2006년도 정부재원(교육인적자원부 학술연구조성사업비)으로 한국학술진흥재단의 지원을 받아 연구되었음(KRF-2006-521-D00437)

1. 서론

최근 지능형 청소 로봇에 관한 연구 및 개발이 활발히 진행되고 있다. 청소 로봇에 관한 연구는 청소 로봇의 청소 모듈 및 몸체를 제작하기 위한 하드웨어 부문, 센서를 중심으로 효율적인 제어를 위한 소프트웨어적인 부분으로 구분할 수 있다. 본 논문에서는 청소 로봇의 움직임에 대한 지능적인 제어를 통해 효율적으로 주어진 공간에 대한 청소를 진행할 수 있는 청소 알고리즘에 관한 새로운 아이디어를 제안한다.

지금까지 청소 로봇을 제어하기 위한 많은 청소 알고리즘들이 연구되어져 왔다(1,2,3,4,5,6,7). 가장 쉬운 접근 방법은 청소 로봇이 주변 환경에 대한 정보를 알지 못한다는 가정 하에 무작위(random) 방식으로 진행되는 알고리즘이 있다. 주변 환경에 대한 정보는 알지 못하지만, 이동 중에 직면하는 장애물의 형태에 따라 효율적인 제어를 시도한다. 그러나 무작위 알고리즘을 적용한 청소 로봇은 청소 대상 공간이 넓어 질수록 청소 시간이 급격히 길어지거나 청소가 되지 않는 공간이 발생하는 등의 문제점을 가지고 있다.

두 번째의 시도는 주변 환경에 대한 정보를 알고 있다는 가정 하에 청소 알고리즘을 설계한다. 가장 기본적인 방법은 일명 바둑판식, 혹은 발걸이 알고리즘으로 알려져 있는 Boustrophedon 경로 기반의 알고리즘이다(1). 이 접근 방법은 전체 청소 구역을 지그재그 방식으로 이동하면서 전체 청소 구역을 이동하는 방식을 취한다. 이 접근 방법은 청소 대상 구역에 장애물이 없거나 적을 경우에는 매우 효율적이거나 장애물이 많을수록 전체 완료 시간이 오래 걸리는 단점이 있다.

지금까지의 로봇 청소 알고리즘의 공통적인 목표는 전체 청소 완료시간의 단축이다. 구체적으로 맵 혹은 주어진 센서를 바탕으로 전체 청소를 완료하는 시간을 단축하고 최적화하는데 초점을 맞추고 있다.

그러나 본 논문에서는 전체 청소 시간의 단축이 아닌 전체 청소 대상 면적의 효율적 순서 매김에 목표를 가지고 있다. 실제 청소 로봇을 활용하다보면 청소 로봇이 전체 청소를 완료할 때 까지를 기다리기가 어려운 경우가 있다. 주어진 시간에 비해서 배터리가 부족해 배터리의 재충전을 기다릴 여유가 없거나, 청소가 완료되기 전에 손님이 방문할 예정이 있거나, 대상 구역에 대한 전체적인 청소보다는 일부 가능한 공간에 대한 약식 청소를 원할 경우 등이 이에 해당될 수 있다. 이러한 경우에는 전체 청소완료시간을 최적화하는 것 보다는 현재 주어진 시간 내에 가장 넓은 면적을 청소하는 것이 유리한 경우가 많다.

이를 위해 본 논문에서는 전체 청소 완료 시간의 단축이 아닌 짧은 시간동안에 가장 효율적으로 가장 넓은 면적을 우선적으로 청소할 수 있는 새로운 로봇 청소 알고리즘을 제안한다.

본 논문의 나머지는 다음과 같이 구성되어진다. 다음절에서는 본 논문에서 제안하는 새로운 청소 알고리즘을 자세히 설명하고, 3절에서 새로운 알고리즘의 동작 과정 및 성능에 대해 분석한다. 4절에서 결론을 맺고 향후 연구 과제를 제시한다.

II. SetClean 알고리즘

이 절에서는 본 논문에서 제안하는 새로운 청소 알고리즘인 SetClean 알고리즘에 대해서 자세히 설명한다.

2.1 공간 분할 방법

청소 로봇이 청소를 진행하기에 가장 효율적인 공간은 장애물이 존재하지 않는 사각형이다. 그러나 실제 공간은 다양한 다각형 혹은 곡선의 모양을 나타낼 수 있으며 물론 다양한 형태의 장애물도 포함할 수 있다. 따라서 새로운 청소 알고리즘의 개발을 위해서 가장 먼저 진행되어야 할 것은 전체 청소 구역을 장애물이 없는 사각형들의 집합으로 재구성하는 일이다.

기존에 청소 대상 구역에 대한 맵이 존재하거나, 혹은 맵이 존재하지 않더라도 최근 슬램 등의 연구의 발전으로 인해 주변 환경에 대한 정보를 미리 혹은 실시간에 획득하는 것은 가능해진 일이다. 이러한 맵을 바탕으로 격자 모양의 그리드 맵을 구성하는 것은 충분히 가능하다. 본 논문의 알고리즘은 그리드 맵을 근거로 진행된다.

그리드 맵에는 청소 대상 구역에 대한 전체 윤곽선이나 장애물의 정보가 포함되어 있다. 가장 먼저 진행할 일은 그리드 맵에서 전체 청소 대상 구역을 장애물이 존재하지 않는 사각형의 집합으로 분할하는 일이다. 이것을 위해 본 논문에서 사용하는 방법은 Rectangle Tiling 기법(8)을 적용하는 것이다.

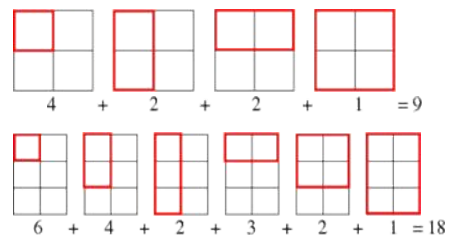


그림 1. Rectangle Tiling 예제(출처: <http://mathworld.wolfram.com/RectangleTiling.html>)
 Fig 1. Example of Rectangle Tiling (source: <http://mathworld.wolfram.com/RectangleTiling.html>)

Rectangle Tiling은 $m \times n$ 의 사각형에서 $N(m,n)$ 의 부분 사각형을 찾아내는 방법이다. 그림 1은 Rectangle Tiling 과정에 대한 한 예를 보여준다. Rectangle Tiling은 한쪽 구석을 기준으로 주어진 왼쪽 아래 코너 (i, j) 를 기준으로 오른쪽 위 코너 $(m-i)(n-j)$ 를 반복해서 선택함으로써 계산되어질 수 있다. 그러므로 $m \times n$ 의 사각형에서 만들 수 있는 부분사각형의 개수는 다음의 식 (1)과 같이 계산할 수 있다.

$$N(m, n) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (m-i)(n-j) = \frac{1}{4} m(m+1)n(n+1). \dots\dots\dots (1)$$

다음 진행해야 할 일은 계산되어진 많은 사각형들 중에 중복되지 않으면서 전체 청소 대상 구역을 포함할 수 있는 사각형을 선택하는 일이다. 이 문제를 해결하기 위해서 본 논문에서 사용하는 방법은 Set Covering 문제를 해결하기 위한 greedy 알고리즘을 적용하는 것이다. Set Covering 문제는 컴퓨터 과학의 알고리즘 분야에서 대표적인 Karp's 21 NP-complete 문제 중의 하나이다. Set Covering 문제는 여러 개의 집합이 입력으로 주어지면, 이 중에서 전체 면적을 포함할 수 있는 최소의 집합들을 선택하는 문제이다.

Set Covering 문제를 polynomial 시간에 해결하기 위한 가장 좋은 방법은 greedy 알고리즘을 적용하는 것이다 [9]. 위의 문제를 해결하기 위한 greedy 알고리즘은 매우 간단하다. 즉, 전체 공간을 커버하기 위한 사각형을 선택하는 각 단계에서 지금까지 커버되지 않은 가장 많은 셀을 포함하는 사각형을 선택하는 것이다.

물론, 각 단계에서 알고리즘을 적용하는 과정에서 Rectangle Tiling을 통해 계산되어진 사각형 들 중 장애물을 포함하는 사각형들은 모두 배제한다. 이러한 과정을 반복하면 전체 공간을 커버하는 최소한의 사각형의 집합을 얻을 수 있게 된다.

2.2 분할 공간 이동 순서 결정 방법

마지막으로 진행되어야 할 단계는 선택된 사각형들에 대한 방문 순서를 결정하는 것이다. 가장 간단한 방법은 서론에서 설명한 바와 같이 가장 넓은 사각형부터 사각형의 면적 순에 따라 차례로 방문하는 것이다.

그러나 실제의 로봇의 청소 과정에서 추가로 고려해야 할 것은 로봇의 현재 위치로부터의 이동 시간 및 회전으로 인한 지연 시간 등이다. 청소 대상 면적의 크기만을 고려하여 이동할 경우 실제 청소 시간보다 로봇의 이동 시간 오버헤드가 더 커질 수 있는 경우도 있다.

또한 동일한 면적이라고 할지라도 사각형의 좌우 길이 비례에 따라 로봇의 회전수가 달라진다. 로봇의 회전이 많을수록 회전 지연으로 인한 청소 시간이 지연될 수 있어 효율성이 달라질 수 있다.

그러므로 효율적인 청소 순서를 결정하기 위해서는 단순히 사각형의 면적뿐만 아니라 로봇의 현재위치로부터의 이동 시간 및 해당 사각형의 청소 과정에서 발생할 수 있는 지연 시간 등을 함께 고려해야 한다.

따라서 본 논문에서는 로봇의 효율적인 이동을 위하여 다음과 같은 단계에 의하여 분할된 사각형들에 대한 방문 순서를 결정한다.

단계 1. 모든 사각형들에 대해 현재 단계에서 청소가 이루어졌을 때 얻을 수 있는 Gain을 계산한다. Gain은 로봇의 현재 위치를 고려하여 해당 사각형을 청소하는데 소요되는 시간에 대한 전체 사각형의 면적의 비율로 결정한다. 임의의 사각형 ri 에 대한 Gain을 구하기 위한 구체적인 공식은 다음과 같다.

$$Gain(ri) = Area(ri) * Timec / ECT(ri)$$

Area(ri): the number of cells in ri

ECT(ri) = Estimated Cleaning Time for ri
 = Distance(L,ri) + Area(ri) * Timec + Nturn(ri) * Timer

Distance(L, ri): the length of path from L to ri

L: current location of robot

Nturn(ri) : the number of turn for cleaning ri

Timec = unit time for cleaning a cell

Timer = unit time for a turn

단계 2. 모든 사각형들 중에서 Gain 값이 가장 높은 사각형을 선택한다. 이를 통해 청소가 이루어지지 않은 사각형 들 중에서 주어진 시간동안에 가장 많은 면적을 청소할 수 있는 사각형을 선택하게 된다.

단계 3. 단계 2를 반복하여 Gain 값이 높은 순서로 청소 순서를 결정한다.

단계 4. 정해진 순서에 따라 청소를 진행한다.

III. 성능 분석

이 절에서는 본 논문에서 제안한 새로운 청소 알고리즘의 동작 과정 및 성능을 분석한다. 알고리즘의 동작 과정을 알아보기 위하여 그림 3과 같은 그리드 맵을 한 예제로 사용한다.

그림 2는 전체 10X10 크기의 셀들로 구성되어 있다. 그림에서 빨간색은 장애물을 의미하며, 진한 파란색 사각형은 로봇의 현재 위치를 나타낸다. 하늘색 사각형 지역들이 청소 대상 구역이 된다.

앞 절에서 제시한 알고리즘에 따라 그림 2의 맵에 대하여 Rectangle Tiling을 실시한 후 Set Covering greedy 알고리즘을 실시한다. 그림 3은 greedy 알고리즘을 수행한 결과를 보여준다. 전체 면적은 R1, R2,.....,R16의 총 16개 공간으로 분할되었으며, R1이 가장 큰 면적의 사각형이고, R16이 가장 작은 면적의 사각형이다. 면적이 동일한 사각형은 왼쪽 위를 기준으로 좌에서 우로, 위에서 아래로 진행하면서 차례로 번호를 부여하였다.

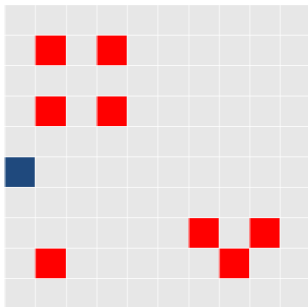


그림 2. 그리드 맵 예제
Fig 2. An Example of Grid Map



그림 3. Set Covering greedy 알고리즘 수행 결과
Fig 3. Result for Set Covering greedy algorithm

그림 4는 그림 3의 결과를 그래프 형태로 표현한 것이다. 그림에서 각 노드는 사각형들을 나타내며, 노드 안의 숫자는 사각형의 면적에 따른 우선순위를 나타낸다. 링크는 한 사각형에서 직면하고 있는 사각형들을 연결해준다.

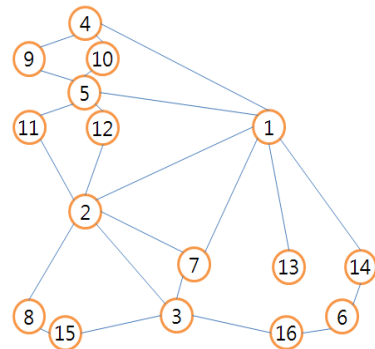


그림 4. 분할된 그리드 맵에 대한 그래프
Fig 4. Graph for Partitioned Grid Map

표 1. 각 단계별 Gain 평가 결과
Table 1. Gain evaluation results for each step

	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16
Cells	42	16	10	4	4	4	2	2	1	1	1	1	1	1	1	1
Move	5	2	6	4	8	5	6	1	1	-	-	4	6	-	1	-
Turns	10	6	2	0	0	2	0	0	0	-	-	0	0	-	0	-
Total	57	24	18	8	12	11	8	3	2	-	-	5	7	-	2	-
Gain	0.74	0.67	0.56	0.5	0.33	0.36	0.25	0.67	0.5	-	-	0.2	0.14	-	0.5	-
Visited	1	5	4	11	9	15	14	6	12	10	8	13	16	2	7	3
Winner																
Via																

이제 진행해야 할 일은 각 노드들에 대하여 방문 순서를 결정하는 일이다. 각 노드에 대한 Gain을 계산해보면 첫 번째 단계에서는 R1이 0.74로 가장 높은 값을 나타낸다. 따라서 첫 번째 단계에서는 R1으로 이동하게 된다. 그림 5를 보면 첫 번째 단계에서의 맵에서 로봇의 이동 내역을 보여준다. 로봇은 s로 표현된 곳에서 출발하여 R1의 in 영역으로 진입한 후 out 영역에서 마무리한다. 이후 동일한 방법으로 나머지 사각형들에 대한 평가 작업을 통해 이동 순서를 결정한다. 그림 6이 최종 단계의 그림을 보여준다.

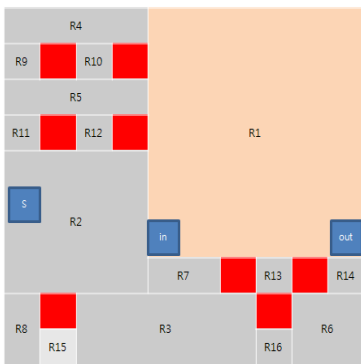


그림 5. 첫 번째 단계
Fig 5. The first step

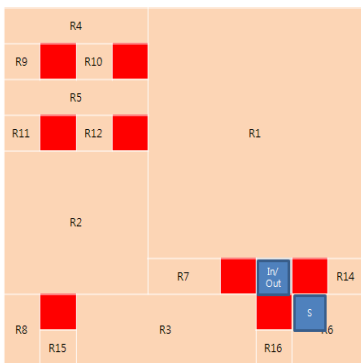


그림 6. 마지막 단계
Fig 6. The last step

표 1은 각 사각형의 셀의 수(cells), 현재 위치로부터 셀까지의 이동 시간(Move), 각 사각형 내에서의 회전 수(Turns), 전체 이동 시간(Total), 각 단계별 예상 청소 완료 시간에 따른 면적 비율(Gain), 방문 순서(Visited) 등을 보여준다. 그림에서 숫자가 없이 '-'로 표현된 부분은 해당 사각

형이 가장 높은 Gain을 보여주어서 선택된 것이 아니라 다른 사각형으로 이동 중에 경유함을 통해 청소가 이루어지는 경우를 말한다. 예를 들어 R1 청소 후 R3로 이동하기 위해서는 R14와 R16을 차례로 경유해야 한다. 그러므로 R1 다음에 R14, R15, 그리고 R3의 순서로 방문 순서가 결정되었다.

그림 7은 시간의 흐름에 따른 청소 완료된 셀의 수를 그래프로 보여준다. 그래프에서 x축은 시간의 흐름을 나타내고, y축은 청소가 완료된 셀의 수를 누적된 형태로 보여준다. 최초 5초간은 로봇의 현재 위치에서 R1으로 이동하는 시간이므로 청소가 진행된 공간이 없고, 이후 57초 동안 R1에 대한 42개의 셀에 대한 청소가 이루어진다. 이후 R3까지의 이동이 다시 y축에서 일정한 값으로 나타난다.

그림 7을 보면 전체적으로 초반 진행 과정에는 많은 면적을 청소하게 되고, 그러면서 청소를 하지 않는 이동시간은 대폭 줄어든다. 반면 후반으로 갈수록 청소하는 시간에 비해 이동시간의 비율이 점차 증가한다.

그림 7의 실험 결과 중에서 흥미로운 것은 전체 청소 완료 시간의 약 30%정도에 약 50%에 해당 하는 면적을, 약 50% 정도의 시간 내에 전체 청소 구역에 대한 약 75%정도의 청소 완료율을 나타내고 있음을 관찰할 수 있다.

이러한 실험 결과를 통해 본 논문에서 제안하는 청소 알고리즘이 전체 청소 구역을 청소하기에 충분한 시간을 갖지 못했을 때 가능한 빠른 시간 내에 많은 면적을 청소할 수 있음을 알 수 있다.

본 논문에서는 제안하는 알고리즘의 일반성 및 실용성을 검증하기 위하여 다양한 실내 공간에 대한 실험을 진행하였으나, 위의 실험 결과와 유사하여 나머지 실험 내용 및 결과는 생략하였다.

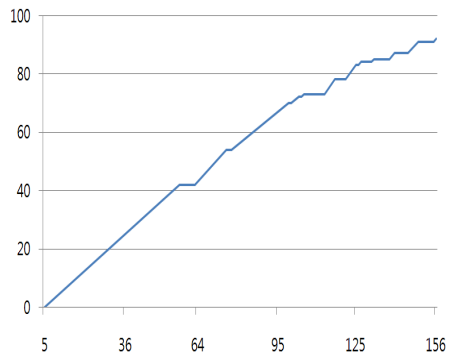


그림 7. 시간의 흐름에 따른 청소 완료된 셀의 수
Fig 7. The number of cleaned cells by time

IV. 결론

본 논문에서는 청소 로봇이 청소를 완료하기에 충분한 시간을 갖지 못하는 상황에서 유용한 새로운 청소 알고리즘을 제시하였다. 새로운 청소 알고리즘은 전체 청소 구역을 청소가 용이한 장애물이 없는 사각형들로 분할 한 후 시간에 비한 청소 면적이 넓은 순서에 의해 전체 면적에 대한 이동 순서를 결정한다.

실험을 통해 본 논문에서 제안하는 알고리즘이 전체 청소 과정의 전반부에 최소한의 이동으로 가능한 많은 공간에 대한 청소를 진행함을 알 수 있다. 구체적으로 주어진 공간에 대해 전체 청소 완료 시간의 50%내에 전체 청소 대상 구역의 75% 이상을 완료하는 등 가능한 빠른 시간 내에 가능한 많은 면적을 청소하였다.

본 논문에서 제안하는 새로운 청소 알고리즘은 전체 대상 공간에 대한 청소를 로봇이 완료하는데 너무 오랜 시간이 걸리거나, 배터리 재충전을 위한 대기 시간이 매우 길 때 등 다양한 상황에서 유용한 실용적인 알고리즘이다.

본 논문에서는 대상 공간 내의 장애물이 모두 정적임을 가정하였으나, 실제 공간에서는 다양한 형태의 동적 장애물이 존재할 수 있다. 따라서 향후 다양한 동적 장애물에 대한 실시간 감지 및 이에 대한 효율적 대처 방안에 대해서 추가 연구를 진행할 계획이다.

참고문헌

- [1] H. Choset and P. Pignon, "Coverage path planning: the boustrophedon cellular decomposition.", In Proceedings of the International Conference on Field and Service Robotics, December 1997.
- [2] A. Zelinsky, R. A. Jarvis, J. C. Byrne, and S. Yuta. "Planning paths of complete coverage of an unstructured environment by a mobile robot.", International Journal of Robotics Research, 13(4), 1994.
- [3] Wesley H. Huang, "Optimal line-sweep-based Decompositions for Coverage Algorithms", In Proceedings of the 2001 IEEE International Conference on Robotics and Automation", 2001.
- [4] E. Acar, H. Choset, "Robust sensor-based coverage of unstructured environments.", In Proceedings of the International Conference on Intelligent Robots and Systems, pp. 61-68, Maui, Hawaii, 2001.
- [5] Yoav Gabriely and Elon Rimon. "Spiral-STC: An on-line coverage algorithm of grid environments by a mobile robot.", In Proceedings of the IEEE International Conference on Robotics and Automation, pages 954-960, Washington, DC, May, 2002.
- [6] Chan Sze Kong, New Ai Peng, and Ioannis Rekleitis, "Distributed Coverage with Multi-Robot System", In Proceedings of the 2006 IEEE International Conference on Robotics and Automation, Orlando, Florida, May 2006.
- [7] E. Garcia and P. Gonzalez de Santos, "Mobile-robot navigation with complete coverage of unstructured environments", Robotics and Autonomous Systems, 46, pp. 195-204, 2004.
- [8] <http://mathworld.wolfram.com/RectangleTiling.html>
- [9] http://en.wikipedia.org/wiki/Set_cover

저 자 소개



전 흥 석

2001년: 홍익대 전자계산학과 박사
 2002년-현재: 건국대 컴퓨터소프트웨어전공 부교수.
 관심분야: 지능형로봇, 임베디드시스템.