

데드라인을 고려하는 효율적인 지능형 로봇 커버리지 알고리즘

전 흥 석^{*} · 정 은 진^{**} · 강 현 규^{*} · 노 삼 혁^{***}

요 약

이 논문은 지능형 로봇을 위한 새로운 커버리지 알고리즘을 제안한다. 커버리지 알고리즘의 성능을 향상하기 위한 많은 연구들은 전체 커버리지 완료 시간을 최소화하는데 초점을 맞추어왔다. 그러나, 만일 전체 커버리지를 완료하기에 충분한 시간이 없다면, 최적의 경로는 달라질 수 있다. 이러한 문제를 해결하기 위하여 본 논문에서는 MaxCoverage라고 하는 데드라인이 있을 경우에 가능한 많은 면적을 커버하기 위한 새로운 커버리지 알고리즘을 제안한다. MaxCoverage 알고리즘은 이동 경로를 셋 커버 문제를 위한 그리디 알고리즘을 이용하여 결정한다. 실험 결과에 의하면 MaxCoverage 알고리즘은 임의의 데드라인에 대하여 다른 알고리즘들에 비해 향상된 성능을 보여준다.

키워드 : 지능형 로봇, 커버리지 알고리즘, 셋 커버, 그리디 알고리즘, 데드라인

An Efficient Coverage Algorithm for Intelligent Robots with Deadline

Heung Seok Jeon^{*} · Eun Jin Jung^{**} · Hyun Kyu Kang^{*} · Sam H. Noh^{***}

ABSTRACT

This paper proposes a new coverage algorithm for intelligent robot. Many algorithms for improving the performance of coverage have been focused on minimizing the total coverage completion time. However, if one does not have enough time to finish the whole coverage, the optimal path could be different. To tackle this problem, we propose a new coverage algorithm, which we call MaxCoverage algorithm, for covering maximal area within the deadline. The MaxCoverage algorithm decides the navigation flow by greedy algorithm for Set Covering Problem. The experimental results show that the MaxCoverage algorithm performs better than other algorithms for random deadlines.

Keywords : Intelligent Robot, Coverage Algorithm, Set Cover, Greedy Algorithm, Deadline

1. 서 론

커버리지 알고리즘은 전체 대상 공간을 최소한 한 번 이상 방문하기 위한 경로 계획을 수립하는 알고리즘이다. 커버리지 알고리즘은 청소 로봇, 지뢰 탐지 로봇, 잔디 깎기 로봇, 페인팅 로봇, 지형 탐색 로봇, 보안 로봇, 농산물 수확 로봇 등 많은 지능형 로봇에서 활용되는 핵심 요소 기술들 중의 하나이다. 본 논문에서는 이와 같이 다양한 커버리지 알고리즘의 응용 분야 중 특히 청소 로봇에 초점을 맞추어 새로운 커버리지 알고리즘을 제안한다.

커버리지 알고리즘의 가장 기본적인 형태는 Boustrophedon 경로를 이용하는 것이다[1]. Boustrophedon 경로는 (그림 1)에서 보여주는 것처럼 일정한 공간을 지그재그 방식으로 교대로 번갈아 왕복하는 알고리즘을 말한다. Boustrophedon 경로는 일명 바둑판 식 알고리즘, 혹은 발갈이 알고리즘 등으로도 알려져 있다.

Boustrophedon 경로 기반 알고리즘은 대상 공간 내에 특별한 장애물이 없거나 비교적 적은 공간에는 매우 효율적인 방법이다. 그러나 장애물이 복잡해질수록 일관적인 방식으로 전체 경로를 계획하는 것이 매우 어려워지며 복잡한 장애물들로 인해 커버리지 성능 또한 급격히 저하된다.

따라서 많은 연구들이 이러한 문제들을 해결하기 위한 새로운 방안들을 제시하고 있다. Huang[2]은 Boustrophedon 경로를 따라 이동 중 로봇의 회전을 최소화하기 위한 방법을 제시한다. Kong et al.[3]은 Boustrophedon 경로를 하나의 로봇이 아닌 여러 로봇으로 주행하는 방법을 제안한다.

* 이 논문은 2008년도 정부재원(교육인적자원부 학술연구조성사업비)으로

한국학술진흥재단의 지원을 받아 연구되었음(KRF-2008-D000413).

† 종신회원: 건국대학교 컴퓨터소프트웨어전공 부교수

** 정 회 원: 아이오와 대학교 컴퓨터과학과 조교수

*** 정 회 원: 홍익대학교 컴퓨터공학과 교수

논문접수: 2008년 9월 3일

수정일: 1차 2008년 11월 12일

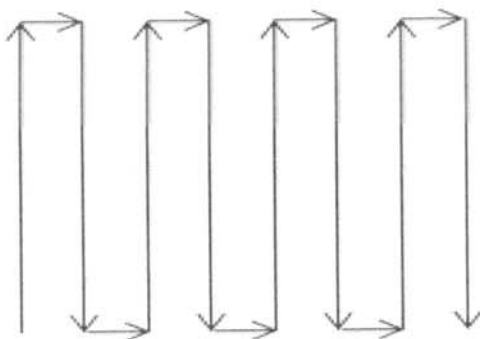
심사완료: 2008년 11월 14일

또한, 기존의 대부분의 커버리지 알고리즘들이 미리 알려진 맵을 기반으로 진행되었으나, 미리 알려지지 않은 장애물들이나 지형 등에 대처하는 알고리즘도 연구되고 있다[4].

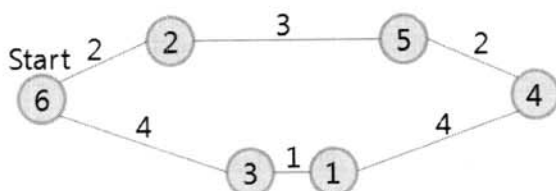
이와 같이 Boustrophedon 경로 방식을 개선하고자 하는 연구들은 다양한 관점에서 커버리지 알고리즘을 개선하고 있다. 그러나 이들 알고리즘들은 궁극적으로 전체 커버리지 시간을 최소화 하고자 하는 공통점을 가지고 있다. 즉, 커버리지의 시작부터 완료될 때까지의 전체적인 관점에서 전체 소요 시간을 최적화하고자 하는 것이다.

그런데, 일상 생활에서는 때때로 예상되는 전체 커버리지 완료 시간을 기다리기에 충분한 시간을 갖지 못하는 경우가 존재한다. 예를 들어, 예상 되는 전체 청소 완료이전에 손님의 방문이 예정된 경우에는 손님이 방문하기 이전에 청소가 마무리되어야 한다. 또한 많은 청소 로봇들이 청소 도중 배터리 전력이 부족할 경우 재충전하는 알고리즘을 수용하고 있다. 그러나 현실적으로 재충전을 하기 위해 소요되는 시간은 결코 짧지 않다. 최소 한 시간에서 서너 시간씩 소요되기도 한다. 이런 경우에 현재 남아 있는 배터리 전력으로 가능한 많은 공간을 청소하는 것이 유용한 경우가 있다.

이때 커버리지가 허용 가능한 시간을 데드라인이라고 한다면 데드라인의 변화에 따라 최적의 이동 경로가 달라질 수 있다. (그림 2)는 데드라인의 변화에 따른 최적 경로가 달라지는 예제를 보여준다. (그림 2)에서 노드는 커버리지의 대상이 되는 부분 영역을 나타내며 링크는 부분 영역들간의 이동 경로를 의미한다. 링크에 해당하는 숫자는 이동 시간을 나타내며, 노드의 숫자는 노드이름, 노드면적, 그리고 노드를 커버하는데 소요되는 시간을 편의상 동일하게 표현하



(그림 1) Boustrophedon 경로



(그림 2) 데드라인에 따라 최적의 커버리지 경로가 달라지는 맵 예제

였다. 그림을 보면 6번 노드에서 시작하여 전체 노드를 모두 방문하는 경우에는 {6,2,5,4,1,3} 순으로 방문하여 33 시간이 소요된다. 그런데, 만일 데드라인이 13이라고 하면 최적의 경로는 {6,3}으로 변경되어야 한다. 또한, 데드라인이 18이라고 하면 최적의 경로는 {6,2,5}의 순으로 다시 변경되어야 한다.

이와 같이 데드라인의 변화에 따른 최적의 경로는 전체 커버리지를 완료하기 위한 최적의 경로와 다를 수 있다.

이러한 문제를 해결하기 위해서 본 논문에서는 기존의 연구들과 달리 전체 커버리지를 완료하기 이전의 임의의 데드라인에 대하여 가능한 많은 면적을 최대한 커버할 수 있는 새로운 커버리지 알고리즘인 *MaxCoverage* 알고리즘을 제안한다. *MaxCoverage* 알고리즘은 전체 영역을 Boustrophedon 경로 방식을 이용하여 방문 가능한 작은 영역들로 분할 한 후 주어진 데드라인에 최대의 면적을 커버할 수 있는 경로 계획을 설정한다. 실험 결과에 의하면 *MaxCoverage* 알고리즘은 다른 알고리즘들에 비해 다양한 임의의 데드라인에 대하여 일관적으로 향상된 성능을 보여준다.

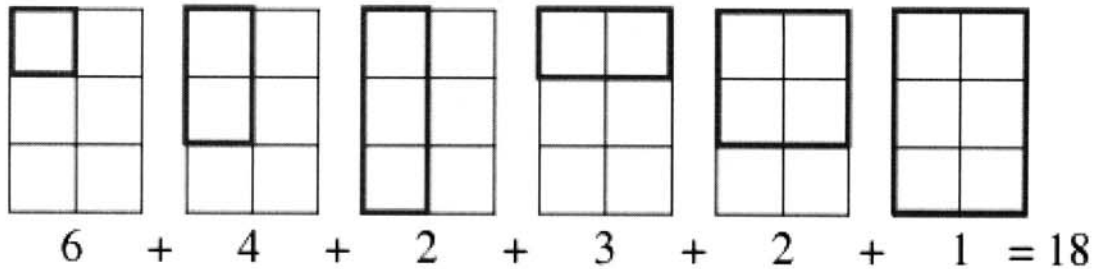
본 논문의 나머지는 다음과 같이 구성된다. 다음 절은 본 논문에서 제안하는 *MaxCoverage* 알고리즘에 대해서 자세히 설명한다. III 절에서는 *MaxCoverage* 알고리즘의 성능을 시뮬레이션을 통해 보여준다. IV 절에서 결론을 맺고 향후 연구 과제를 제시한다.

2. MaxCoverage 알고리즘

임의의 데드라인에 대해 최대의 면적을 커버하기 위해서는 주어진 시간 동안에 최대의 효율을 발휘할 수 있는 공간을 우선적으로 처리해야 한다. 이를 위해서 고려해야 할 세 가지 기본적인 원칙들이 있다.

첫 번째는 전체 면적 대비 장애물이 상대적으로 덜 복잡한 지역을 우선적으로 방문 하는 것이 유리하다. 로봇의 이동 시간 대비 최고의 방문 효율을 높이기 위해서는 모든 지역을 중복 방문 없이 한번씩만 방문하는 것이 가장 좋은 방법이다. 그러나 일반적인 실내 공간에는 테이블, 쇼파, 의자 등의 장애물들로 인해 부분적으로 다양한 형태의 장애물들이 존재한다. 이러한 장애물이 복잡하게 존재하는 공간에서는 로봇의 센서를 통한 장애물 감지 처리 및 회전으로 인한 지체와 이미 방문했던 지역에 대한 중복 방문의 가능성이 높다. 즉, 장애물이 많을수록 장애물의 회피를 위해 요구되는 오버헤드가 크기 때문에 시간당 방문 효율을 높이기 위해서는 상대적으로 장애물이 적은 지역을 먼저 방문하는 것이 좋은 방법이다.

둘째는 해당 공간 내에서의 로봇의 회전이 적은 공간을 먼저 커버하는 것이 유리하다. 앞서 설명한 것처럼 장애물이 존재하는 공간에서는 당연히 이의 회피를 위하여 로봇의 회전이 필요하지만, 장애물이 존재하지 않는 공간에서도 진행 방향 끝에서 벽 혹은 제한 구역을 만났을 경우 회전이 필요하다. 그런데, 동일한 면적의 공간이라도 전체적인 지

(그림 3) Rectangle Tiling 예제(출처: <http://mathworld.wolfram.com/RectangleTiling.html>)

형에 따라 회전의 수가 달라진다. 예를 들어 동일한 면적의 정사각형과 직사각형을 고려해보자. 만일 로봇이 직사각형 내에서 긴 축의 방향으로 이동한다고 가정했을 경우에, 직사각형보다는 정사각형의 공간이 더 많은 로봇의 회전을 요구하게 된다. 이러한 회전으로 인해 동일한 면적이라도 정사각형에 대한 전체 커버리지 완료 시간이 더 늦어지는 것이다. 그러므로 로봇의 방문 효율을 높이기 위해서는 로봇의 회전이 적게 발생하는 공간을 우선적으로 방문해야 한다.

마지막으로, 로봇의 현재 위치를 고려해야 한다. 장애물이 존재하지 않는 공간이고, 로봇의 회전 수가 적은 공간이라고 할 지라도 로봇의 현재 위치와 멀리 떨어진 공간이라면 해당 공간까지의 이동 시간이 추가로 요구된다. 그러므로 효율적인 방문 지역을 선택하기 위해서는 로봇의 현재 위치로부터의 상대적인 이동 거리를 고려해야 한다.

본 논문에서는 이상과 같은 세 가지 기본적인 원칙을 기반으로 새로운 커버리지 알고리즘을 설계한다. 새로운 알고리즘의 설계에 앞서서 몇 가지 다음과 같은 가정을 한다. 먼저 본 논문에서는 다른 커버리지 알고리즘과 관련한 많은 연구들과 동일하게 그리드 방식의 맵을 기반으로 진행된다. 기존에 커버리지 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[5, 6, 7].

앞 절에서 설명한 것처럼 Boustrophedon 경로 방식의 커버리지 알고리즘은 장애물이 존재하지 않을 경우에는 매우 효율적인 방식이다. 그러나 실 생활에서의 우리의 실내 주거 공간은 다양한 형태의 장애물이 포함할 수 있다. 따라서 본 논문에서는 전체 공간을 Boustrophedon 경로 방식을 이용할 수 있는 장애물이 없는 부분 영역들로 분할한다. 구체적으로 설명하면 다음과 같다.

그리드 맵에서 전체 청소 대상 구역을 장애물이 존재하지 않는 사각형의 집합으로 분할하기 위해서 Rectangle Tiling 기법[8]을 사용한다. Rectangle Tiling은 격자로 이루어진 사각형에서 구성 가능한 부분 사각형들을 찾아내는 수학적 기법이다. $m \times n$ 의 격자로 이루어진 사각형의 경우 Rectangle Tiling을 통해 생성할 수 있는 부분 사각형의 개수는 예를 들어 왼쪽 아래 코너를 기준으로 오른쪽 위 코너 $(m-i)(n-j)$ 를 반복해서 선택함으로써 계산될 수 있다. 이를 공식화

하면 $m \times n$ 의 사각형에서 만들 수 있는 부분 사각형의 개수는 다음의 식과 같다.

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

(그림 3)은 Rectangle Tiling 과정의 한 예제를 보여준다. (그림 3)에서 3×2 의 사각형으로서 가능한 부분 사각형은 모두 18개가 존재한다.

이와 같이 Rectangle Tiling을 통해서 전체 공간을 분할하기 위한 부분 공간들의 후보들을 생성할 수 있다.

다음 진행해야 할 일은 Rectangle Tiling을 통해 계산된 많은 사각형들 중에 중복되지 않으면서 전체 커버리지 대상 구역을 포함할 수 있는 사각형들을 선택하고 그 사각형들간의 방문 순서를 결정하는 일이다. 여러 사각형들 중에서 전체 공간을 포함하는 최소의 사각형을 구하는 문제는 Set Covering 문제와 동일한 성격의 문제이다. Set Covering 문제는 컴퓨터 과학의 알고리즘 분야에서 대표적인 Karp's 21 NP Complete 문제들 중의 하나이다[9]. Set Covering 문제는 전체 공간에 대한 여러 개의 부분 집합들이 입력으로 주어지면, 이 중에서 전체 면적을 포함할 수 있는 최소의 집합들을 선택하는 문제이다. 즉, Rectangle Tiling을 통해 도출된 사각형들의 집합 중에서 전체 커버리지 대상 공간을 포함하는 최소의 사각형들을 선택하는 것이다.

Set Covering 문제를 해결하기 위한 좋은 방법 중의 하나는 그리드 알고리즘을 적용하는 것이다 [10]. 이 문제에서 그리드 알고리즘을 적용하는 방법은 매우 간단하다. 즉, 각 단계에서 주어진 시간 동안에 방문 우선 순위가 가장 높은 사각형을 선택하는 것이다. 방문 우선 순위는 방문 시간에 대한 방문 면적의 비율로 정의할 수 있다. 구체적으로 설명하자면, Rectangle Tiling을 통해 도출된 사각형들 중에서 현재까지 선택되지 않은 지역만을 포함하는 사각형을 추출하여, 해당 사각형을 모두 커버하는 시간에 비한 사각형의 면적의 비율을 구하여, 그 중에서 가장 큰 값을 선택하는 것이다.

마지막으로 진행되어야 할 것은 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 이동 과정에서 고려해야 할 것은 로봇의 현재 위치로부터의 이동 시간을 고려해야 한다. 청소 대상 면적의 크기만을 고려하여 이동할 경우 실제 방문 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 커버리지 순서를 결정하기 위해서는 단순히 사각형의 면적뿐만 아니라 로봇의 현재 위치로부터의 이동 시간 및 해당 사각형의 방문 과정에서 발생할 수 있는 회전 지연 시간 등을 함께 고려해야 한다.

그런데, 이때 고려해야 할 중요한 사항이 바로 테드라인이다. 위에서 언급한 사항들을 고려하여 해당 부분 영역들을 방문하되, 각각의 부분 영역을 방문 하기 전에 주어진 테드라인 내에 완료할 수 있는지를 검토해야 한다. 만일 테드라인 내에 해당 부분 영역에 대한 방문을 완료할 수 있다면 그대로 진행하면 되지만, 주어진 테드라인 내에 완료할 수 없다면, 다음 순서의 부분 영역들을 검토하여 테드라인 내에 만족할 수 있는 사각형을 다시 선택해야 한다.

지금까지 설명한 일련의 과정을 알고리즘으로 정리하면 (그림 4)와 같다.

Algorithm *MaxCoverage*:

Step 1: Rectangle Tiling을 이용하여 전체 공간에 대한 부분 사각형을 추출한다.

Step 2: 추출된 사각형들의 시간대비 면적의 비율이 높은 순으로 선택하여 전체 면적을 포함하는 최소의 사각형 집합을 선택한다.

Step 3: 로봇의 현재 위치를 고려하여 선택된 모든 사각형들에 대한 예상 커버리지 완료시간을 계산한다.

Step 4: 각 사각형들의 예상 커버리지 완료 시간 대비 사각형 면적의 비율이 가장 높은 사각형을 선택한다.

Step 5: 선택된 사각형이 테드라인을 만족하는지 검사한다. 만족하지 못하면 Step 4를 반복하여 새로운 사각형을 선택한다.

Step 6: 모든 사각형들이 선택될 때까지 Step 3 ~ 5를 반복한다.

Step 7: 선택된 사각형들의 순서대로 이동한다. 각 사각형 내에서는 Boustrophedon 경로 방식을 이용하여 전 영역을 방문한다.

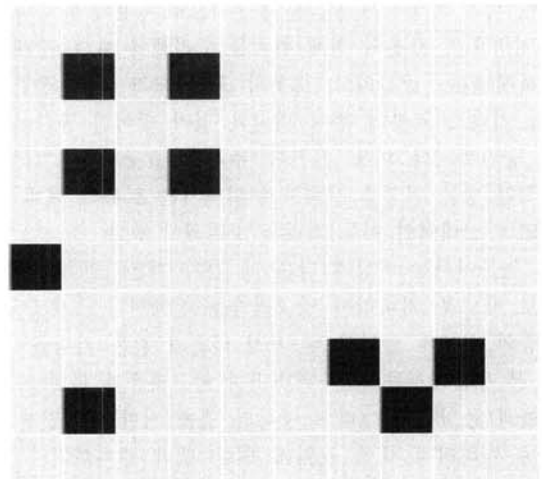
(그림 4) *MaxCoverage* 알고리즘

3. 성능 분석

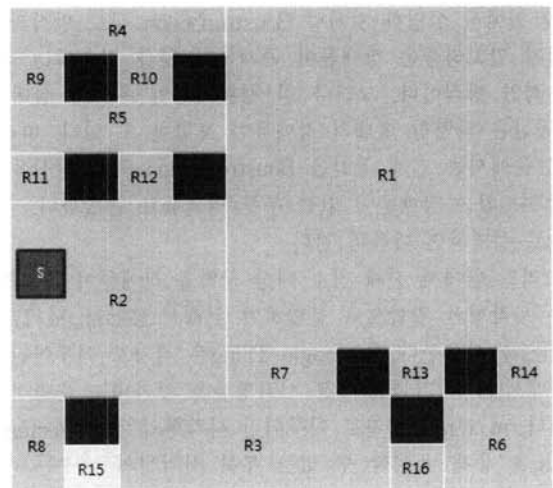
이 절에서는 본 논문에서 제안한 새로운 커버리지 알고리즘의 동작 과정 및 성능을 분석한다. 알고리즘의 동작 과정을 알아보기 위하여 (그림 5)와 같은 그리드 맵을 예제로 사용한다.

(그림 5)는 전체 10x10 크기의 셀들로 구성되어 있다. 그림에서 빨간색은 장애물을 의미하며, 진한 파란색 사각형은 로봇의 현재 위치를 나타낸다. 하늘색 사각형 지역들이 커버리지 대상 구역이 된다.

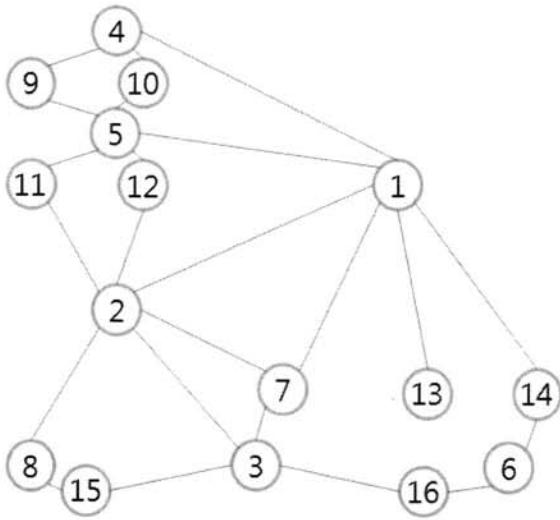
앞 절에서 제시한 *MaxCoverage* 알고리즘에 따라 (그림 5)의 맵에 대하여 Rectangle Tiling을 실시한 후 Set Covering을 위한 그리디 알고리즘을 실시한다. (그림 6)은 *MaxCoverage* 알고리즘을 통한 영역 분할 결과를 보여준다. 전체 면적은 {R1, R2,.....,R16}의 총 16개 공간으로 분할되었다. 면적이 동일한 사각형은 왼쪽 위를 기준으로 좌에서 우로, 위에서 아래로 진행하면서 차례로 번호를 부여하였다.



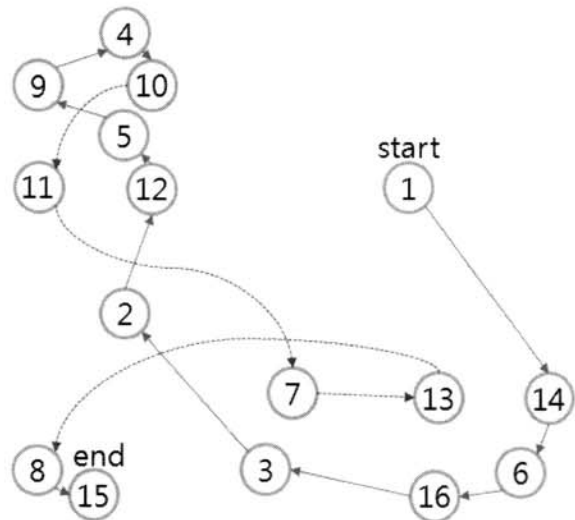
(그림 5) 성능 평가를 위한 그리드 맵 예제



(그림 6) *MaxCoverage* 알고리즘을 통한 영역 분할 결과



(그림 7) 분할된 그리드 맵에 대한 그래프



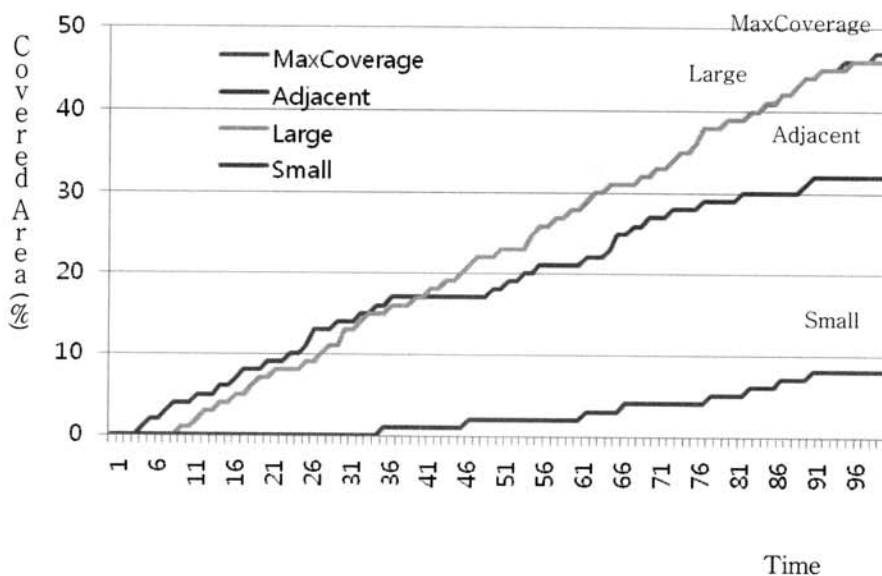
(그림 8) MaxCoverage 알고리즘에 의한 방문 순서도

(그림 7)은 (그림 6)의 결과를 그래프 형태로 표현한 것이다. 그림에서 각 노드는 사각형들을 나타내며, 링크는 한 사각형에서 직면하고 있는 다른 사각형들과의 연결 관계를 표현한다.

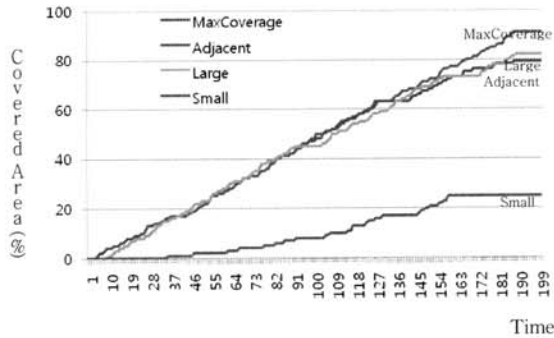
(그림 8)은 데드라인이 없는 상태에서의 MaxCoverage 알고리즘에 의한 노드들 간의 방문 순서를 보여준다. 그림에서 파란색 실선은 인접한 노드들 간의 이동을 보여주며, 빨간색 점선은 인접하지 않은 노드들 간의 이동을 보여준다. 그림을 보면 전반부에는 인접한 면적이 넓은 사각형들을 중심으로 이동한 후 후반부에 이르러 인접하지 않은 작은 사

각형들을 방문하고 있음을 알 수 있다.

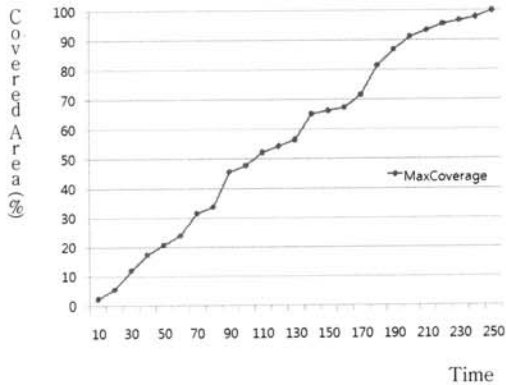
본 논문에서는 MaxCoverage 알고리즘과의 성능 비교를 위해 Adjacent, Large, 그리고 Small 등의 세 가지 다른 알고리즘을 함께 구현하여 실험하였다. 지금까지 커버리지를 위해 발표된 알고리즘은 많이 있지만 [2,3,4] 본 연구와 유사하게 전체 공간을 분할하고, 분할한 공간에 대한 경로 계획을 세우는 적절한 비교대상이 존재하지 않아 데드라인을 고려하는 것의 효율성을 검증하기 위하여 데드라인을 고려하지 않는 단편적인 세 극단의 알고리즘을 비교 대상으로 선정하였다. 먼저 Adjacent 알고리즘은 가장 큰 면적에서 시



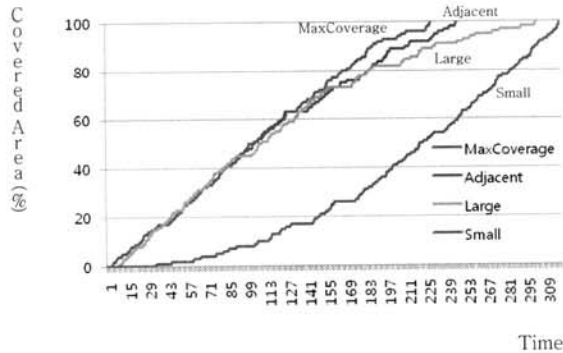
(그림 9) 데드라인 100인 경우의 시간의 흐름에 따른 커버리지 면적의 누적 비율



(그림 10) 데드라인 200인 경우의 시간의 흐름에 따른 커버리지 면적의 누적 비율



(그림 11) 데드라인의 변화에 대한 MaxCoverage 알고리즘의 적응성



(그림 12) 데드라인이 없을 경우의 시간의 흐름에 따른 커버리지 면적의 누적 비율

작하여 가장 인접한 사각형들을 차례로 방문하는 알고리즘이다. 만일 인접한 사각형이 여러 개일 경우에는 더 넓은 면적의 사각형을 선택한다. 따라서 *Adjacent* 알고리즘에 의한 방문 순서는 {2,1,4,9,5,10,11,12,7,3,15,8,16,6,14,13}과 같다.

Large 알고리즘은 사각형의 면적이 큰 순서대로 방문하

는 알고리즘이고, *Small* 알고리즘은 사각형의 면적이 작은 순서대로 방문하는 알고리즘이다.

(그림 9)는 데드라인이 100인 경우의 시간의 흐름에 따른 커버리지 면적의 누적된 비율을 각각의 알고리즘 별로 그래프로 보여준다. 그래프에서 x축은 시간의 흐름을 나타내고, y축은 청소가 완료된 셀의 수를 누적된 형태로 보여준다.

데드라인 100의 의미는 전체 커버리지 시간의 약 40%에 해당하는 시간이다. 그래프를 보면, *Adjacent* 알고리즘의 경우 초반에 *MaxCoverage* 알고리즘에 비해 좋은 결과를 보여주나 중반부터 이동시간이 많아 결과적으로 *MaxCoverage* 알고리즘에 비해 떨어지게 된다.

Large 알고리즘과 *MaxCoverage* 알고리즘은 비슷한 성능을 나타내다 데드라인과 인접해서 *MaxCoverage* 알고리즘의 성능이 조금 높게 나타난다. 적은 면적부터 이동하는 *Small*은 전체적으로 다른 알고리즘에 비해 성능이 많이 떨어지는 결과를 보여준다.

(그림 10)은 데드라인이 200인 경우의 시간의 흐름에 따른 커버리지 면적의 누적된 비율을 보여준다. 그림을 보면 초반에는 *MaxCoverage*, *Adjacent*, 그리고 *Large* 세 알고리즘들이 비슷한 성능을 보여주지만, 후반부에 이르러 *MaxCoverage* 알고리즘이 다른 알고리즘들에 비해 앞서는 결과를 보여준다.

(그림 9)와 (그림 10)을 분석해보면, 데드라인이 짧을 때와 길 때 모두 *MaxCoverage* 알고리즘이 다른 알고리즘들에 비해 좋은 성능을 보여줄 수 있다. 이것은 *MaxCoverage* 알고리즘이 다양한 형태의 데드라인을 잘 고려하여 적응해나가기 때문이다. 이러한 결과는 (그림 11)에서도 확인할 수 있다. (그림 11)은 시간 간격 10 단위의 데드라인의 변화에 따라 *MaxCoverage* 알고리즘의 누적된 커버리지 면적의 비율을 보여준다. 그림을 보면 *MaxCoverage* 알고리즘은 Deadline의 변화에 따라 특별한 굴곡 없이 선형적으로 증가함을 알 수 있다. 이를 통해 *MaxCoverage* 알고리즘이 다양한 형태의 데드라인에 관계없이 잘 적응해나감을 확인할 수 있다.

이러한 특성으로 인해 *MaxCoverage* 알고리즘이 데드라인이 주어질 경우에 최대의 면적을 커버하기 위한 목적으로 설계되었지만, 데드라인이 주어지지 않은 상황에서도 좋은 결과를 나타낸다. (그림 12)를 보면 데드라인이 없는 경우에 전체 면적을 커버하는 과정을 보여준다. 다른 세 알고리즘에 비해서 *MaxCoverage* 알고리즘이 가장 빠른 시간에 커버리지를 완료한다. 이를 통해 *MaxCoverage* 알고리즘이 다양한 데드라인에 대해서 일관되게 좋은 성능을 가져올 수 있음을 확인할 수 있다.

4. 결 론

본 논문에서는 로봇이 커버리지를 완료하기에 충분한 시

간을 갖지 못하는 상황에서 유용한 새로운 커버리지 알고리즘인 *MaxCoverage* 알고리즘을 제안하였다. *MaxCoverage* 알고리즘은 Rectangle Tiling을 통해 전체 영역을 장애물이 없는 부분 영역들로 분할하고, Set Covering 문제를 해결하기 위한 그리디 알고리즘을 적용하여 데드라인 내에 최대의 공간을 방문할 수 있도록 이동 경로를 계획한다.

MaxCoverage 알고리즘은 임의의 데드라인에 대해서 다른 알고리즘들에 비해 좋은 성능을 나타내었으며, 데드라인이 없는 경우에도 좋은 성능을 보여주었다.

본 논문에서는 장애물이 모두 정적임을 가정하였으나, 실제 공간에서는 다양한 형태의 동적 장애물 및 변화하는 정적 장애물들이 존재한다. 따라서 향후 실시간으로 주변 환경에 대한 감지 및 이에 대한 효율적 대처 방안에 대해서 추가로 연구를 진행할 계획이다.

참 고 문 헌

[1] H. Choset and P. Pignon, "Coverage Path Planning: the boustrophedon cellular decomposition", In Proceedings of the International Conference on Field and Service Robotics, Dec., 1997.

[2] Wesley H. Huang, "Optimal Line sweep based Decompositions for Coverage Algorithms", In Proceedings of the 2001 IEEE International Conference on Robotics and Automation, 2001.

[3] 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.

[4] 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.

[5] G. Dissanayake, P. Newman, H. F. Durrant Whyte, S. Clark, and M. Csobra, "A Solution to the Simultaneous Localisation and Mapping (SLAM) Problem", IEEE Transactions on Robotics and Automation., Vol.17, No.3, pp.229-241, 2001.

[6] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "Fast SLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem", In Proceedings of AAAI National Conference on Artificial Intelligence, 2002, pp.593-598.

[7] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "Fast SLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Probably Converges", In Proceedings of International Joint Conference on Artificial Intelligence, pp.1151-1156,

2003.

[8] Ian Stewart, "Squaring the Square", Scientific American, pp.74-76, July, 1997.

[9] Richard M. Karp, "Reducibility Among Combinatorial Problems", Complexity of Computer Computations, R. E. Miller, J. W. Thatcher, Eds., Plenum Press, 1972.

[10] Uriel Feige, "A Threshold of Inn for Approximating Set Cover", Journal of the ACM (JACM), Vol.45 No.4, pp.634-652, July, 1998.

전 흥 석



e-mail : hsjeon@kku.ac.kr
 1996년 홍익대학교 컴퓨터공학과(학사)
 1998년 홍익대학교 전자계산학과(석사)
 2001년 홍익대학교 전자계산학과(박사)
 2002년~2006년 건국대학교

컴퓨터소프트웨어전공 조교수

2007년~현 재 건국대학교 컴퓨터소프트웨어전공 부교수
 관심분야: 지능형 로봇, 임베디드시스템, 운영체제

정 은 진



e-mail : ejjung@cs.uiowa.edu
 1999년 서울대학교 컴퓨터공학부(학사)
 2002년 텍사스 오스틴 대학교 대학원
 전산학과(석사)
 2006년 텍사스 오스틴 대학교 대학원
 전산학과(박사)

2006년~현 재 아이오와 대학교 컴퓨터과학과 조교수
 관심분야: 컴퓨터보안, 센서네트워크

강 현 규



e-mail : hkkang@kku.ac.kr
 1985년 홍익대학교 전자계산학과(학사)
 1987년 한국과학기술원 전산학과(석사)
 1992년 정보처리기술사 자격 취득
 1997년 한국과학기술원 전산학과(박사)
 1987년~2000년 한국전자통신연구원
 책임연구원

2001년~2005년 건국대학교 컴퓨터소프트웨어전공 조교수
 2006년~현 재 건국대학교 컴퓨터소프트웨어전공 부교수
 관심분야: 정보 검색, 자연어처리, 한국어정보처리, 모바일
 컴퓨팅 등



노삼혁

e-mail : samhnoh@hongik.ac.kr

1986년 서울대학교 컴퓨터공학과(학사)

1993년 매릴랜드 대학교 컴퓨터과학과(박사)

1994년~현재 홍익대학교 컴퓨터공학과
교수

관심분야: 운영체제, 임베디드시스템, 실시간 시스템, 병렬 및
분산시스템