

자율 지능형 로봇을 위한 그룹화 기반의 효율적 커버리지 알고리즘

전 흥 석*, 노 삼 혁**

Efficient Coverage Algorithm based-on Grouping for Autonomous Intelligent Robots

Heung Seok Jeon *, Sam H. Noh **

요 약

최근 슬램 알고리즘의 실현을 통해 주변 환경에 대한 맵 정보가 획득 가능할 경우에 격자 그리드 기반의 Boustrophedon 경로 기반 커버리지 알고리즘이 매우 효율적인 것으로 알려져 있다. 그러나 Boustrophedon 경로 기반 알고리즘은 실내 공간에 장애물이 복잡하게 존재할 경우에는 급격히 성능 저하 현상이 발생한다. 따라서 본 논문에서는 복잡한 실내 공간에서도 효율적으로 빠른 시간 내에 청소를 완료할 수 있는 Group-k 알고리즘을 제안하고 구현한다. Group-k 알고리즘은 전체 공간을 장애물의 복잡성에 근거하여 전체 공간을 그룹화하고 각 그룹별 우선순위를 부여하여 전체 작업 순서를 효율적으로 제어한다. 구현 기반의 실험에 의하면, 본 논문에서 제안된 알고리즘은 Boustrophedon 경로 기반 알고리즘에 비해 약 20%의 성능 향상을 보여준다.

Abstract

The coverage algorithm based on Boustrophedon path has been known to be the most efficient in places without or less obstacles. If the map of an environmental area thru SLAM algorithm can be obtainable. However, the efficiency of the coverage algorithm based on Boustrophedon path drops drastically when obstacles are complex. In this paper, we propose and implement a new algorithm, which we call Group-k, that efficiently handles the complex area. The Group-k algorithm groups the obstacles and prioritizes the covering sequences with complex rank of the groups. Implementation-based experiments show about 20% improved performance when applying the new algorithm, compared to the Boustrophedon path algorithm.

▶ Keyword : 지능형로봇(Intelligent robot), 커버리지(coverage), 복잡성(complex rank)

• 제1저자 : 전흥석

• 접수일 : 2008. 1. 24, 심사일 : 2008. 2. 15, 심사완료일 : 2008. 2. 20.

* 건국대학교 컴퓨터소프트웨어전공 부교수, **홍익대학교 컴퓨터공학전공 교수

※ 본 연구는 산학협동재단 학술연구비 (0601020) 지원으로 수행되었음.

I. 서론

청소 로봇은 다른 지능형 로봇에 비해서 가장 많이 현실화 되어가고 있는 대표적인 차세대 지능형 로봇 중의 하나이다. 그러나 청소로봇은 기존의 진공청소기를 대체하여 사람의 노동을 완전히 대신하지는 못하고 있다. 이러한 현상의 이유 중의 하나는 커버리지 알고리즘의 비효율성 때문이다. 현재 출시되어있는 제품에 대한 커버리지 알고리즘을 분석해보면 대부분이 랜덤 (random) 방식의 알고리즘에 기반을 두고 몇 가지 형태의 패턴으로 진행되는 커버리지 알고리즘을 적용하고 있다. 이러한 방식의 커버리지 알고리즘은 설계 및 구현이 용이하다는 장점이 있지만 청소 시간이 매우 오래 걸리는 단점을 가지고 있다.

이러한 문제를 해결하기 위하여 최근 청소 로봇에 Boustrophedon 경로[1] 방식을 적용한 커버리지 알고리즘들에 대한 연구들이 많이 진행되고 있다. Boustrophedon 경로 기반 커버리지 알고리즘은 청소 대상을 격자 방식으로 왕래 하면서 청소하기 때문에 랜덤 방식의 알고리즘에 비해 획기적인 성능 향상을 가져오게 된다.

그러나 Boustrophedon 경로 방식을 적용하기 위해서는 로봇이 주변 환경에 대한 맵 정보를 미리 알고 있어야 한다는 제약 조건이 있다. 주변 환경에 대한 정보를 알아야 지그재그식의 알고리즘을 효율적으로 구동시킬 수 있기 때문이다.

이러한 지능형 로봇의 맵 정보 구축은 최근 많이 연구되고 있는 슬램 (SLAM: Simultaneous Localization And Mapping)에 관한 연구를 통하여 해결할 수 있다. 슬램은 로봇의 위치 인식과 맵 정보 구축을 동시에 진행하기 위한 연구로써 최근 이론적 해결 방안 및 시범적 연구들이 많은 긍정적인 결과를 보여주고 있다.

그런데, Boustrophedon 경로 기반의 커버리지 알고리즘은 장애물이 없거나 적은 공간에서는 좋은 성능을 보여주지만 장애물이 많아 복잡한 공간에 대한 청소 시에는 장애물 처리를 위한 지연으로 인해 성능 저하 현상이 발생한다(2,3,4). 실제로 청소 대상이 되는 일반적인 실내 구조는 매우 비정형적이고 복잡한 특징을 가지고 있다. 우리의 주거 공간은 그 크기나 구조가 매우 다양하다. 더욱이 실내 구조에 배치된 식탁, 쇼파, 테이블, 의자 등의 여러 장애물들의 배치는 유동적일 수 있어 그 어려움을 더하게 된다. 더욱이 벽면, 구석, 낮은 장애물의 아래 부분, 장애물의 윗부분 등에 대해서는 로봇 스스로 완벽히 청소를 진행할 수가 없다. 이러한 부분 등에 대해서는 사람이 마무리를 해주어야 만족할 만한 청결 수준에

도달할 수 있다 [5].

따라서 청소 로봇이 사람의 손을 전적으로 대신하여 모든 것을 청소해주는 것을 기대하기 보다는 사람과 로봇의 적절한 협력 시스템을 갖추는 것이 더 실용적일 수 있다. 예를 들어 청소 로봇이 장애물이 적은 부분을 먼저 청소하고, 사람이 장애물이 복잡한 공간을 이후에 혹은 동시에 마무리하면 청소 로봇이 청소를 완료하는데 소요되는 실질 청소 완료시간은 매우 많이 단축될 수 있다. 따라서 청소 로봇이 사람과의 협력을 전제하여 효율을 높이기 위해서는 가능한 빠른 시간 내에 전체 대상 공간 중에서 복잡하지 않은 많은 부분에 대해 청소를 완료하는 것이 바람직하다.

따라서 이러한 문제를 해결하기 위하여 본 논문연구팀에서는 사람과 로봇의 협력을 통해서 효율적인 청소를 진행할 수 있는 새로운 커버리지 알고리즘인 Group-k 알고리즘을 제안하였다. 간단히 설명하면 Group-k 알고리즘에서는 전체 청소 대상 공간을 장애물들의 복잡성의 정도에 의해 그룹들을 구성하고, 전체 그룹들 중에서 복잡하지 않은 그룹부터 로봇이 청소를 진행한다. 이를 통해 전체 청소 구간에 대해 가능한 빠른 시간 내에 많은 면적이 청소 되어 인간과 상호 협력을 통해 청소 작업을 빠른 시간 내에 완료할 수 있다.

본 논문에서는 Group-k 알고리즘을 실제의 로봇에 구현하여 기존의 Boustrophedon 경로 기반의 커버리지 알고리즘과 성능을 비교 분석한다.

II. 관련 연구

청소 로봇이 효율적으로 전체 영역을 커버하기 위해서는 주변 환경에 대한 맵 정보를 미리 알고 있어야 한다. 이와 같이 실시간으로 로봇의 위치 인식 및 주변 환경에 대한 맵 정보를 구축하기 위한 슬램에 관한 대표적인 연구는 확장 칼만 필터를 이용한 EKF-SLAM(6)과 파티클 필터 기반의 FastSLAM 알고리즘[7, 8] 등이 있다.

EKF-SLAM 알고리즘은 부가적인 가우시안 노이즈와 함께 상태-공간 모델의 형식으로 관찰 모델과 모션 모델을 표현하는데, 슬램 문제를 해결하기 위해 확장형 칼만 필터 (EKF)를 사용한다. EKF-SLAM은 SLAM에 관한 가장 기본적인 알고리즘으로써 이후의 SLAM 연구에 많은 영향을 끼치고 있다. 그러나 EKF-SLAM의 가장 큰 문제는 실제 적용을 위한 과정에서 계산 오버헤드가 너무 크다는 점이다. 관찰 갱신 단계에서 랜드마크의 수가 증가함에 따라 계산의 과정이 급격하게 증가한다는 것이다. 또한 EKF-SLAM은 로봇 포즈의 높은 불확실성으로 인해 데이터 연관 (data association) 작

업을 실패할 확률이 높다는 단점을 가지고 있다.

또 다른 중요한 대안은 로봇의 모션 모델을 좀 더 보편적인 비-가우시안 확률 분포의 샘플들의 집합으로 묘사하는 FastSLAM 알고리즘이다. FastSLAM 알고리즘의 주요 아이디어는 Rao-Blackwellization(R-B)을 적용하여 샘플 공간을 줄이는데 있다. FastSLAM은 랜드마크의 수가 K 이고, 파티클의 수가 M 일 경우에 복잡도 $O(M \log K)$ 를 가진다. 그러나 FastSLAM 알고리즘은 주변에 충분한 파티클이 존재하지 않을 경우에 loop closure에 어려움을 나타낼 수 있다. 그럼에도 불구하고, FastSLAM 알고리즘은 실제 여러 로봇에서 비교적 정확한 맵을 생성할 수 있음을 보여준다[7, 8].

III. Group-k 알고리즘

본 논문에서는 기존의 연구들과 달리 사람과 로봇의 협력을 통해 현실적인 커버리지 시간을 단축하기 위한 새로운 알고리즘인 Group-k 알고리즘을 제안한다. 사람과 로봇이 협력하기 위해서는 각각의 특성을 잘 살려야 한다. 로봇은 장애물이 적고 넓은 공간을 비교적 빠른 시간에 이동할 수가 있지만, 장애물이 복잡한 공간에서는 어려움이 많으며 완벽한 청소를 기대하기가 어렵다. 반면에 사람은 복잡한 공간에 대한 섬세한 청소를 잘 할 수 있다. 그러므로 복잡한 공간은 로봇이 청소를 하고, 복잡하지 않은 공간은 사람이 청소를 하는 것이 바람직하다.

구체적으로 전체 공간을 복잡한 정도에 따라 분류하여, 로봇은 가장 복잡하지 않은 공간부터 청소를 진행하고, 사람은 가장 복잡한 공간부터 청소를 시작하면 두 주체가 만나는 시점에서 청소가 완료될 수 있다. 이때 가능한 빠른 시간 내에 로봇이 많은 면적을 청소 완료하게 되면 복잡한 공간부터 청소를 진행하는 사람의 노동력을 최소화할 수 있다.

3.1 장애물의 그룹화

장애물들을 그룹화하기 위하여 본 논문에서는 Nearest Neighbour 알고리즘을 이용한다. Nearest Neighbour 알고리즘은 원래 Traveling Salesperson Problem을 해결하기 위한 알고리즘으로서 본 연구에 적용하기 위해서는 약간의 수정이 필요하다. 구체적으로, Nearest Neighbour 알고리즘은 전체 노드의 방문 순서를 결정하기 위하여 시작 노드로부터 가장 인접한 노드를 선택하고, 이후의 선택 대상에서 이미 선택되었던 노드들을 제외하여 선택함으로써 전체 노드를 하나의 링크로 연결하는 과정이다. 그러나 본 논문에서는 그

룹을 형성하기 위하여 각 장애물로부터 가장 인접한 노드들을 선택하여 가장 인접한 노드들 간에 링크를 생성한다. 즉, 모든 노드에 대한 최 인접 노드의 선택 대상에 기존의 선택 여부와 관계없이 모든 노드들이 대상이 된다. 모든 노드에 대한 가장 인접한 노드를 선택하게 되면 링크를 통해 연결된 노드들을 하나의 그룹으로 설정한다. 이러한 과정을 알고리즘으로 정리하면 다음과 같다.

Algorithm Grouping_Obstacles:

- step 1. Make two sets of nodes, set A and B , and make a set of groups, set G
- step 2. Put all nodes into set A and B
- step 3. Pick a arbitrary node a in set A
- step 4. Search the node b in set B which is the closest to the node a , and make the link ab between the nodes a and b
- step 5. **If** (a group g_i which has the link containing a or b node exists in set G) **then**
append the link ab to the group g_i
- Else**
generate new group g_{new}
append the link ab to the group g_{new}
- step 6. Repeat step 3 until all nodes are in set A is empty.

3.2 그룹의 복잡성

Grouping_Obstacles 알고리즘에 의하여 장애물의 그룹화를 진행하면 많은 인접한 장애물들이 각각의 그룹으로 형성된다. 하지만 Grouping_Obstacles 알고리즘에 의한 장애물들의 그룹화 결과가 항상 적절한 것은 아니다. 때로는 최 인접 노드끼리는 하지만, 두 노드를 포함한 그룹이 전체 면적을 모두 차지하거나, 많은 복잡한 노드들을 하나의 그룹으로 집약하여 장애물들을 그룹화 하는 의미를 상실할 수도 있다. 그러므로 Grouping_Obstacles 알고리즘을 통한 결과를 최종적인 그룹으로 사용하기 위해서는 그 적절성을 판단하고 조정하는 단계가 필요하다.

이 문제를 해결하기 위하여 본 논문에서는 그룹의 복잡성 ($C(g_i)$: Complex Rank of group i) 이라는 개념을 도입한다.

그룹의 복잡성을 정의할 수 있는 방법에는 여러 가지가 존재할 수 있겠지만, 본 논문에서는 히스토그램 방법을 사용한다[9]. 히스토그램은 영상의 명암 값 프로파일을 보여주지 위

해 사용되는 도구이다. 히스토그램은 영상의 전체 명암 값의 분포를 담을 수 있는 도구로서 이를 이용하면 영상의 명암도 분포 상태를 알 수 있다. 구체적으로 히스토그램 구성 방법은 0부터 255까지의 명암 값을 인덱스로 하고, 영상을 구성하고 있는 각 화소의 명암 값에 해당하는 개수를 빈도수로 간주하여 1씩 증가시키는 방법을 사용한다. 이때 0은 검은색을 의미하고, 255는 흰색을 나타낸다.

실내 구조도를 히스토그램으로 표현하는 방법은 실내에 존재하는 장애물을 검은색 픽셀, 청소 대상 공간이 되는 바닥을 흰색 픽셀들의 집합으로 가정하여 처리한다. 그러면 전체 실내 구조도는 흰색과 검은색 등 두 색의 분포로 이루어지게 되며 이때 검은색 혹은 흰색에 대한 픽셀 값의 분포도를 측정하여 이미지의 복잡도를 판단할 수 있다. 픽셀에 대한 색상 분포 히스토그램을 분석하면 평균값이나 표준 편차 등의 영상에 관한 다양한 정보를 정량적으로 분석할 수 있는데 이 값들 중에서 본 논문에서 의미가 있는 것은 평균값(mean)이다. 평균값이 크면 클수록 이미지 상에 분포되어 있는 검은색 픽셀 수가 적은 것이며, 값이 적을수록 검은색 값이 많이 분포한다는 의미이다. 따라서 평균값이 작을수록 실내 공간이 복잡한 구조임을 나타내고, 값이 클수록 단순한 구조라고 판단할 수 있다 [9].

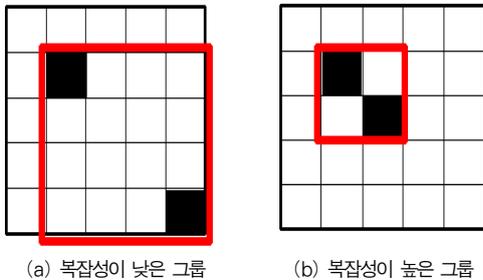


그림 1. 복잡성 판단의 적절한 예
Fig. 1 An appropriate example of deciding complex rank based on histogram.

히스토그램에 의하여 복잡성을 판단할 경우 그림 1과 같이 동일한 장애물이라도 그룹의 전체 면적이 다를 경우에는 평균값이 달라지기 때문에 두 그룹의 복잡성이 달라져서 히스토그램의 평균값이 그룹의 적정성을 판단하는 타당성 있는 기준임을 알 수 있다.

그러나 히스토그램을 이용한 복잡성의 분석은 전체 대상 그룹의 면적과 그룹 내의 장애물의 전체 면적은 동일하나, 그 구조가 다를 경우에 문제점이 발생한다. 예를 들어 그림 2와 같은 경우를 살펴보자. 그림 2를 보면 직관적으로 당연히 (a)

) 보다는 (b)의 구조가 더 복잡함을 알 수 있다. 그러나 히스토그램은 두 가지 경우 모두 동일한 색상 분포 값을 나타내게 된다. 이것은 히스토그램이 장애물의 지리적 위치나 모양은 전혀 고려하지 않기 때문에 발생하는 문제이다. 그러므로 장애물의 복잡성을 측정하기 위해 히스토그램에 의한 색상의 분포만을 고려하면 합리적이지 못한 판단을 할 수 있다.

따라서 이를 보완하기 위하여 히스토그램의 평균값 이외에 장애물의 외양적 구조를 추가로 고려하여 그룹의 장애물 복잡성 산출 과정을 개선한다. 장애물의 외양적 구조는 장애물이 동일한 면적 분포를 차지하지만 모양이 다를 경우를 구별하기 위한 것이다.

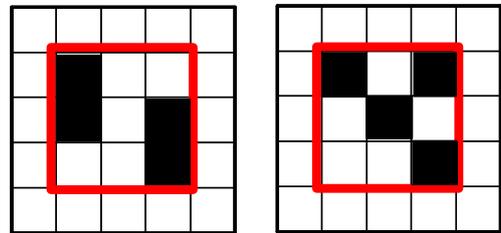


그림 2. 복잡성 판단의 부적절한 예
Fig. 2 An inappropriate example of deciding complex rank based on histogram.

장애물의 외양적 구조를 나타내는 방법은 여러 가지가 가능할 수 있겠지만 본 논문에서는 그룹 내의 장애물들의 전체 꼭짓점의 수에 따라 장애물의 구조적인 복잡성을 판단한다. 즉, 꼭짓점이 많은 다각형일수록 복잡하다고 판단하고, 적을수록 단순한 구조로 판단하는 것이다. 원 모양의 장애물도 세 부적으로는 사각형 모양의 단위 셀에 맵핑 되므로 복잡한 다각형의 구조로 인식된다. 장애물의 꼭짓점을 고려할 경우 그림 2에서 왼쪽의 (a) 그림은 전체 8개의 꼭짓점을 가지고, 오른쪽 (b)의 그림은 13개의 꼭짓점을 가지게 되므로 오른쪽 그림이 왼쪽 그림에 비해 더 복잡한 것으로 판단할 수 있는 근거를 제공해준다.

정리하면, 본 논문에서는 그룹의 복잡성을 히스토그램의 평균값과 장애물의 외양적 구조를 나타내는 꼭짓점의 수를 조합하여 판단한다. 문제는 서로 다른 측정 단위인 히스토그램의 평균값과 장애물의 꼭짓점의 수를 어떻게 결합할 것인가이다. 이를 위해 본 논문에서는 각각의 두 요소들을 최소 값이 0이고 최대 값이 1이 되는 정규화된 비율로 변환하여 두 비율의 평균값을 복잡성으로 결정한다. 이때 복잡성 값 0의 의미는 아무런 장애물이 없는 상태이고, 1은 가장 복잡한 구조임

을 나타낸다.

예를 들어 특정 그룹 (g_i)에 대한 히스토그램의 평균값 ($H_{mean}(g_i)$)의 경우 0에서 255사이의 값을 산출하게 되는데 이를 0에서 1사이의 비율로 변환한다. 히스토그램의 경우에 0은 검은색을 나타내어 가장 복잡한 경우를 나타내고, 255는 흰색을 나타내어 가장 단순한 경우를 나타내므로 히스토그램의 평균값이 0일 경우를 1, 255일 경우를 0으로 변환한다.

둘째, 장애물의 꼭짓점의 수는 그룹 내 가능한 꼭짓점 수의 합 ($A_{sum}(g_i)$)에 대한 그룹 (g_i)내 장애물들의 꼭짓점 수의 합 ($A_{obstacle}(g_i)$)의 비율로써 산정한다. 만일 그룹 내 전체 꼭짓점 수의 합과 장애물의 꼭짓점 수의 합이 동일할 경우를 1로 산정하고, 그룹 내의 장애물이 없는 경우는 0이 된다. 그룹 내 장애물의 꼭짓점 수가 전체 꼭짓점 수의 반이 될 때의 비율은 0.5가 된다.

따라서 특정 그룹 (g_i)의 복잡성 $C(g_i)$ 는 위에서 언급한 두 요소들의 평균값으로 결정되며 이러한 내용을 수식으로 정리하면 다음과 같다.

$$C(g_i) = \frac{(1 - \frac{H_{mean}(g_i)}{255}) + (\frac{A_{obstacle}(g_i)}{A_{sum}(g_i)})}{2} \quad (1)$$

3.3 그룹의 해제 및 통합

다음 단계는 그룹의 복잡성 C 를 이용해서 전체 그룹들을 조절하는 것이다. 그룹 조절 작업은 크게 두 단계로 이루어진다. 첫 번째 단계는 그룹의 복잡성이 특정한 기준 k 이하일 경우 해당 그룹 내의 장애물에 대한 그룹화가 적절하지 못하다고 판단하여 해당 그룹을 축소 혹은 해제하는 것이다. 이를 위한 구체적인 방법은 다음과 같다.

먼저 실내 공간의 전체 그룹에 대해 순차적으로 각 그룹별 복잡성을 계산한다. 만일 특정 그룹의 복잡성이 k 이하일 경우에는 해당 그룹 내에서 가장 긴 링크를 해제하고 남은 장애물들로 그룹을 재형성하여 그룹의 복잡성을 다시 계산한다. 장애물들의 그룹을 형성하는 과정에서 각 장애물들은 전체 장애물들 중에서 가장 인접한 노드를 선택하도록 하였기 때문에 가장 긴 링크를 해제하면 하나의 노드, 즉 하나의 장애물과 나머지 장애물들의 그룹으로 분리된다. 이러한 과정을 그룹의 복잡성이 k 이상이 될 때까지 모든 그룹에 대해 반복한다. 만약 한 그룹 내의 모든 링크가 해제되면 해당 그룹 자체가 해제된다. *Release_Group* 과정을 거치고 나면 남은 모든 그룹들의 복잡성은 적어도 k 이상이 된다.

Algorithm *Release_Group*:

- step 1. Compute the complexity $C(g_i)$ of group g_i
- step 2. **If** ($C(g_i) < k$) **then**
Release the longest link in the group g_i
- step 3. Repeat Step 1 until $C(g_i) > k$
- step 4. Repeat Step 1 for all the groups in G

다음 단계는 그룹의 범위를 확장 조절하는 것이다. 이것은 생성된 소규모 그룹의 수가 너무 많을 경우 전체 면적에서 그룹에 속하지 않은 부분의 구조가 복잡해지는 것을 방지하기 위한 것이다. 이 과정도 첫 번째 단계의 *Grouping_Obstacles* 알고리즘과 유사하게 Nearest Neighbour 알고리즘을 응용한다. 먼저 각 그룹별 가장 인접한 그룹을 선택하여 두 그룹을 가상으로 병합한 후 병합한 가상 그룹의 복잡성을 계산한다. 만일 가상 그룹의 복잡성이 k 이상이 되면 두 그룹을 실제로 합병하게 된다. 이 과정은 다음의 *Merge_Group* 알고리즘에서 보여준다.

Algorithm *Merge_Group*:

- step 1. Select the Nearest Neighbour group of the group g_i in G
- step 2. Make a virtual group v of g_i and its nearest neighbour group
- step 3. Compute the complexity $C(v)$ of group v
- step 4. **If** ($C(v) \geq k$) **then**
Merge the group g_i and its nearest neighbour group
- Else**
Release the group v
- step 5. Repeat step 1 for all the groups in G

3.4 커버리지 알고리즘

실내 공간의 장애물들에 대한 그룹 조절 작업이 모두 이루어지면 이제 생성된 그룹을 기반으로 청소를 해야 할 차례이다. 청소는 모든 그룹들의 복잡성을 계산하여 이를 바탕으로 복잡성이 낮은 순서대로 정렬하는 작업부터 진행된다. 청소는 먼저 그룹에 속하지 않은 전체 구역을 먼저 청소하고, 그 다음부터는 복잡성이 낮은 그룹 순서대로 청소를 실시한다. 만약 복잡성이 동일하다면 이전 청소한 그룹과 가장 가까운 거리에 있는 그룹부터 청소를 한다. 물론 그룹 내에는 바둑판식 알고리즘을 적용하여 청소를 한다. 이러한 과정을 정리한 것이 다음 알고리즘이다.

Algorithm *Group-k*:

- step 1. Build a map for the space to be cleaned and

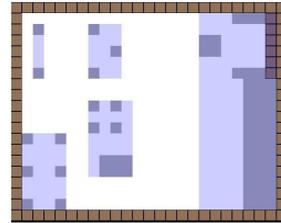
- Initialize the position of the cleaning robot
- step 2. Call the procedure *Grouping_Obstacles*
- step 3. Call the procedure *Release_Group*
- step 4. Call the procedure *Merge_Group*
- step 5. Sort and make a list L for all the groups by the complexity $C(g_i)$ in increasing order
- step 6. Clean the area which has not been included in any group.
- step 7. Pick the group g_i from the head of the list L
- step 8. Move the cleaning robot to the group g_i by the shortest path search algorithm
- step 9. Clean the group g_i by the plowing_method algorithm
- step 10. Repeat step 7 until all groups are in list L is empty.

IV. 성능 평가

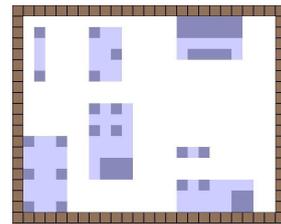
이 절에서는 본 논문에서 제안한 Group- k 커버리지 알고리즘의 구현 내용 및 성능 평가 결과를 분석한다.

Group- k 알고리즘의 구현을 위해 (주)한울로보틱스 사의 Hanuri-RD 모델 로봇을 사용하였다. Hanuri-RD는 임베디드 리눅스 기반으로 초음파 센서 12개와 카메라 센서 등이 설치되어있는 연구용 로봇이다. 청소 로봇을 위한 알고리즘의 성능을 평가하긴 하지만 실제 청소 모듈을 설치하지는 않았다. 실제 청소 모듈의 성능과 본 논문의 알고리즘의 성능과는 독립적인 문제이므로 실험의 편의를 위하여 그리드 맵을 기반으로 로봇이 지나간 셀은 청소가 완료되었다고 판단한다.

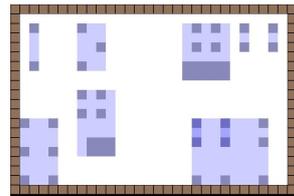
청소 대상이 되는 주변 환경에 대한 맵 정보 구축 및 로봇의 위치 인식은 FastSLAM 알고리즘[14]를 적용하였다. 청소를 시작하기 전 일정 시간동안 FastSLAM 슬랩 알고리즘을 통하여 주변 환경에 대한 맵 정보를 구축하고, 이 정보를 바탕으로 격자 그리드 모양의 맵으로 변환작업을 진행하였다. 해당 셀에 장애물이 존재할 경우 해당 셀은 점유 상태로 표현한다. 그림 3은 실험을 진행한 공간에 대한 격자 그리드 모양의 맵을 보여준다. 그림에서 짙은 파란색 부분은 장애물이 존재하는 셀을 나타내며, 하늘색 사각형은 Group- k 알고리즘을 적용하여 얻어진 각각의 그룹을 나타낸다.



(a) 실험 공간 1



(b) 실험 공간 2



(c) 실험 공간 3

그림 3. 실험공간에 대한 격자 그리드 맵 및 그룹화 결과.
Fig. 3 Grid maps and grouping results for each experimental area.

그림 4는 그림 3의 각각의 실험 공간에 대한 실험 결과를 보여준다. 그림의 X 축은 시간의 흐름을 나타내며, 그림의 y 축은 전체 공간에 대한 청소 완료 비율을 의미한다. 그러므로 그래프는 시간의 흐름에 따른 청소 완료 비율을 나타낸다. 그래프 중에서 분홍색 그래프는 Group- k 알고리즘을 적용한 경우를 보여주며, 파란색 그래프는 Boustrophedon 경로 기반 알고리즘을 적용한 결과를 보여준다.

그림 4의 결과를 보면 4가지 실험 모두 Group- k 알고리즘의 청소 완료 시간이 그룹핑을 하지 않은 경우에 비해 평균 약 20% 정도 단축되었음을 알 수 있다. 이것은 동일한 장애물이라도 Group- k 알고리즘은 지역적으로 분포하는 장애물들을 그룹화 하여 이동의 반경을 제한하지만 바둑판식의 경우는 그 제한이 없이 이동 경로 상의 장애물들을 폭 넓게 처리하기 때문에 전체 지체 시간이 더 많이 발생하게 된다. 이것은 Boustrophedon 경로 기반 알고리즘의 청소 완료율 곡선이 장애물 처리를 위해 여러 번 계단식 분포를 나타내는 것을 보면 알 수 있다.

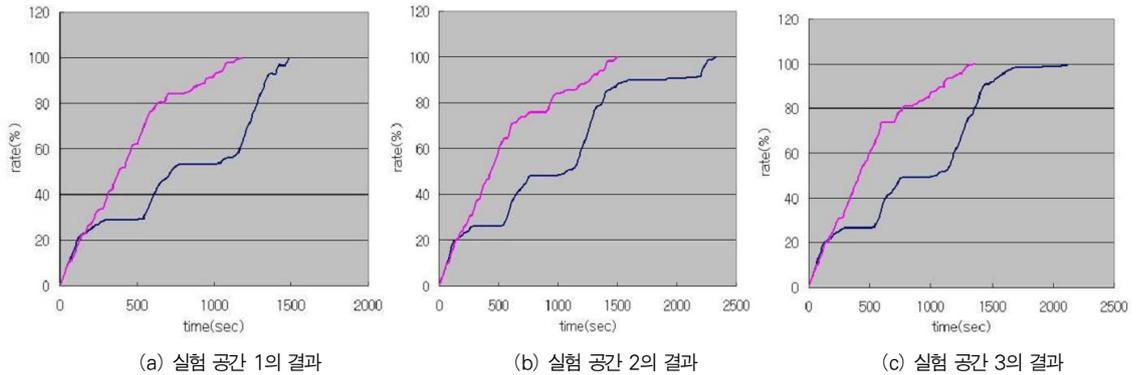


그림 4. 시간의 흐름에 따른 커버리지 영역의 변화.
Fig. 4 Accumulated variation of covered area by time.

주목해야할 것은 위의 실험 결과는 사람의 협력 없이 로봇 스스로 모든 청소를 완료했을 경우의 결과이다. 본 논문의 서론에서 설명한 것과 같이 복잡하지 않은 공간을 로봇이 청소할 동안 사람이 복잡한 공간을 함께 청소할 경우 전체 청소 완료시간은 훨씬 더 단축된다. 가령 로봇이 전체 청소 구역의 80% 정도를 청소할 동안 사람이 복잡한 공간부터 청소하여 전체 청소 구역의 약 20%정도에 대한 청소를 진행한다라고 가정한다면, 실질적인 청소 완료 시간은 청소 완료율 80% 시점에서 완료될 것이다. 이것은 바둑판식 청소알고리즘에 비해 평균 약 50%정도의 성능 향상을 가져오게 된다.

이처럼 사람과 로봇이 효율화된 시스템으로 잘 협력한다면 지능형 로봇의 서비스가 인간의 삶에 미치는 긍정적인 효과는 실질적으로 증가할 것이다.

V. 결론 및 향후 연구

본 논문에서는 지능형 청소 로봇과 사람의 협력을 통해서 실질적인 청소 완료 시간을 단축할 수 있는 새로운 커버리지 알고리즘인 Group-k 알고리즘을 제안하였다. 기존의 Boustrophedon 경로 기반의 커버리지 알고리즘은 장애물이 적은 공간에서는 효율적이거나 장애물이 많아 복잡한 공간에서는 청소 완료 시간이 길어지는 문제점을 가지고 있다. 또한 실제 실내 공간에서는 장애물의 특성 및 위치에 따라 로봇이 청소하기에는 매우 어려운 부분들이 존재한다. 이러한 문제를 해결하기위하여 Group-k 알고리즘은 전체 공간을 장애물의 복잡성에 따라 그룹으로 형성하고 복잡성이 낮은 순서에 따라 청소를 진행한다. 이러한 특성을 이용하여 복잡한 공간을 사람이 함께 청소한다면 실질적인 커버리지 완료 시간을 급격히 줄일 수 있다.

Group-k 알고리즘은 슬램 알고리즘을 통하여 청소 대상이

되는 실내 공간의 구조 및 정적 장애물에 대한 정보를 파악할 수 있고 이를 바탕으로 효율적인 주행 계획을 수립할 수 있다. 그러나 사람과 같은 실내에 존재하는 동적인 장애물의 출현 시 이에 대한 고려를 하지 못하고 있다. 그러므로 동적 장애물이 많이 존재하는 실제 생활공간에서의 활용을 위하여 동적 장애물에 대한 감지 및 대응 방안에 대한 연구가 추가로 이루어져야 할 것이다.

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] G. Schmidt and C. Hofner, "An Advanced Planning and Navigation Approach for Autonomous Cleaning Robot Operations", In Proceedings of the IEEE Int. Symposium on Intelligent Vehicles, pp.364-369, 1995.
- [3] I. Ullrich, F. Mondada and J.-D. Nicoud, "Autonomous Vacuum Cleaner", Robotics and Autonomous Systems, vol. 19, pp.233-245, 1997.
- [4] R. Neumann de Carvalho, H. A. Vidal, P. Viera and M. I. Riberio, "Complete Coverage Path Planning Guidance for Cleaning Robots", In Proceedings of the IEEE Int. Symposium on Industrial Electronics, vol. 2, 1997.
- [5] 오연택, "청소로봇의 소비자 조사", In Proceedings of 2nd Annual Workshop of Korea Robotics Society, 2005.

- [6] G. Dissanayake, P. Newman, H. F. Durrant-Whyte, S. Clark, and M. Csobza, "A Solution to the Simultaneous Localisation and Mapping (SLAM) Problem", IEEE Transactions on Robot. Automat., vol. 17, no. 3, pp.229-241, 2001.
- [7] 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, pp.593-598, 2002.
- [8] 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 Int. Joint Conference on Artificial. Intelligence, pp. 1151-1156, 2003.
- [9] Lee, Hyok-Jun and Lee, Jong-Suk, "Study of Digital Analysis Efficiency through a Complexity Analysis", Journal of Korea Institute of Interior Design, vol 31, 2002.

저 자 소 개



전 흥 석
2001년: 홍익대 전자계산학과 박사
2002년-현재: 건국대 컴퓨터소프트웨어전공 부교수.
관심분야: 지능형 로봇, 임베디드 시스템.



노 삼 혁
1993년: 메릴랜드대학교 컴퓨터과학과 박사
1994년 - 현재: 홍익대학교 정보컴퓨터공학부 교수.
관심분야: 운영체제, 시스템 소프트웨어, 임베디드 시스템