

## 실내 공간의 복잡성을 고려한 실용적 청소 알고리즘

전흥석+ 조재욱\* 노상혁\* 나대영+<sup>o</sup>

+건국대학교 컴퓨터응용과학부

\*홍익대학교 컴퓨터공학과

hsjeon@kku.ac.kr, mrtajo@cs.hongik.ac.kr, samhnoh@hongik.ac.kr, sorinara@kku.ac.kr<sup>o</sup>

### Practical Cleaning Algorithm based on Complex Rank of Indoor Environment

Heung Seok Jeon+, Jaewook Jo\*, Sam H. Noh\*, D.Y. Na+<sup>o</sup>

+Dept. of Computer Science, Konkuk University

\*Dept. of Computer Engineering, Hongik University

#### 요 약

청소로봇은 대부분 랜덤방식 또는 바둑판식의 청소알고리즘으로 운용된다. 랜덤 알고리즘은 전체 청소 시간이 오래 걸린다는 단점을 가지고 있다. 랜덤 알고리즘의 문제를 해결하기 위한 바둑판식 알고리즘은 현재까지 가장 좋은 알고리즘으로 알려져 있으나 장애물이 복잡한 공간에서는 청소시간이 길어지는 단점을 가지고 있다. 이런 문제점을 해결하기 위하여 본 논문에서는 Group-k 라는 새로운 청소 알고리즘을 제안한다. Group-k 알고리즘은 청소시간을 단축시키는 목적보다는 청소시간은 같더라도 복잡한 구역일수록 나중에 청소함으로써 가능한 빠른 시간 내에 가장 많은 면적을 청소하는 것을 목표로 한다. 즉 인접한 복잡한 장애물들을 하나의 그룹으로 구성하고 그룹의 복잡성을 계산하여 복잡성이 낮은 그룹부터 먼저 청소하는 방식이다. 시뮬레이션에 기반한 실험을 통해 Group-k 알고리즘이 복잡한 장애물 구역을 그룹화하여 복잡한 공간을 효율적으로 청소함을 보여준다.

#### 1. 서 론

현재까지 출시된 청소 알고리즘에 대하여 분석하여 보면 랜덤 방식의 알고리즘을 가진 청소로봇은 청소시간이 매우 오래 걸리는 단점을 가지고 있다. 이러한 문제점을 획기적으로 개선한 바둑판식 알고리즘은 장애물이 없거나 적은 공간에서는 좋은 성능을 보여준다. 그러나 장애물이 많거나 복잡한 공간에 대한 청소 시에는 장애물 처리를 위한 지연으로 인해 성능 저하 현상이 발생한다 [2,3,4]. 청소로봇의 청소 대상이 되는 실내의 구조는 비정형적이며 동적인 특징을 가진다. 또한, 청소로봇의 구매자를 생각하면 직장여성이 청소로봇을 선호할 것이라는 생각과는 달리 실질적으로 주부가 많이 구매한 사실을 주목해 볼 필요가 있다. 이것은 청소로봇과 사람이 함께 같은 공간에 있을 가능성이 높은 것을 나타낸다. 따라서 청소 로봇이 사람의 손을 전적으로 대신하여 모든 것을 청소해주는 것을 기대하기 보다는 사람과 로봇이 적절한 협력 시스템을 구축하는 것이 실용적일 수 있다. 청소로봇이 장애물이 적은 공간을 청소하고 사람은 장애물이 많은 복잡한 공간을 청소하여 마무리하는 것이 더 효율적일 수 있다. 즉, 청소를 완료하는데 소요되는 전체 시간은 동일하다할 지라도 빠른 시간 내에 많은 부분이 청소 완료되는 것이 바람직하다.

이를 위해 본 논문에서는 전체 청소 공간에서 복잡성에 따라서 그룹을 형성하고, 복잡하지 않은 그룹부터 청소함으로써 전체 청소 구간에 대해 가능한 빠른 시간 내에 많은 면적이 청소 되어 인간과 상호 협력을 통해 청소 작업을 완료할 수 있는 Group-k 청소 알고리즘을 제안한다.

#### 2. 관련 연구

지금까지 바둑판식 알고리즘의 장애물 처리에 관한 몇 가지 형태의 개선 시도가 이루어지고 있다 [4,5,6,7,8,9]. 대표적으로 전체 공간을 장애물이 적은 n개의 공간으로 나누어 개별 공간에 바둑판식 알고리즘을 적용한 후 다시 연결 하는 것이다. [4,5].

이와 유사한 것으로 전체 공간을 그래프를 적용하여 각 노드를 최단 방문 경로 해법을 이용하는 해법을 이용하는 연구도 이루어지고 있다[10].

이러한 연구들의 공통적인 목표는 바둑판식 알고리즘을 개선하여 청소 완료시간을 단축하는 것이다. 즉, 이 연구들은 전체 청소시간을 단축하는 것을 목표로 하고 있다.

그러나 본 논문에서는 기존의 관련된 연구와는 달리, 주어진 공간에서 얼마나 빠른 시간 내에 많은 공간을 청소하느냐에 초점을 맞추고 있다. 가능한 빠른 시간 내에 장애물이 적은 공간이 먼저 청소가 완료되고 사람은 복잡한 공간을 청소하여 청소로봇이 복잡한 공간을 청소하는 것을 최대한으로 줄임으로, 사람과 로봇의 협동으로 전체적인 청소 완료 시간을 단축시킬 수 있을 것이다.

#### 3. Group-k 알고리즘

##### 3.1 장애물들의 그룹화

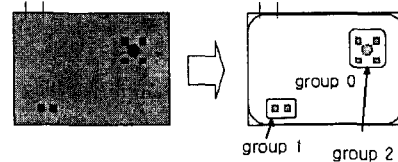


그림 1 장애물의 그룹화

인접한 장애물들을 그룹화 하기위하여 본 논문에서는 Nearest Neighbour 알고리즘을 수정하여 이용한다. Nearest Neighbour 알고리즘은 전체 방문순서를 결정하기 위하여 시작노드를 선택하고 이후의 선택 대상에서 이미 선택되었던 노드들을 제외하여 선택하나, 본 논문에서는 그룹을 형성하기 위하여 각 장애물로부터 가장 인접한 노드들 간에 링크를 생성하도록 한다. 즉, 모든 노드들이 선택에 상관없이 링크를 연결하는 대상이 되는 것이다. 이 과정에서 가장 인접한 노드가 둘 이상일 경우에는 해당 노드 모두에 대해 링크를 생성한다. 모든 노드들을 하나의 그룹으로 설정하게 되면 연결된 링크를 통해 하나의 그룹으로 설정한다. 자세한 내역은 그림 2에 제시한다.

##### Algorithm Basic\_Group:

- 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 closest to the node a, and make the link ab between the nodes a and b
- step 5. If (the group gi which has the link containing a or b node exists in set G) then  
append the link ab to the group gi  
Else  
generate new group gnew  
append the link ab to the group gnew
- step 6. Repeat step 3 until all nodes are in set A is empty.

그림. 2 Basic\_Group 알고리즘

3.2 Basic\_Group 알고리즘의 문제점

Basic\_Group 알고리즘에 의해 장애물들의 그룹화를 진행하면 많은 장애물들을 그룹으로 형성된다. 그러나 생성된 그룹이 항상 적절한 그룹은 아닐 수 있다. 예를 들어 그림 3과 같은 경우를 고려하면 a, b, c를 장애물이라고 보면 직관적으로 a, b를 하나의 그룹으로, c는 그룹에서 해제되는 것이 바람직하다. 하지만 Basic\_Group 알고리즘에 의하면 a, b, c 모두를 하나의 그룹으로 묶어진다. 그러므로 Basic\_Group 알고리즘을 통한 결과를 최종적인 그룹으로 사용하기에는 그룹의 적절성을 판단하고 조정하는 단계가 필요하다.

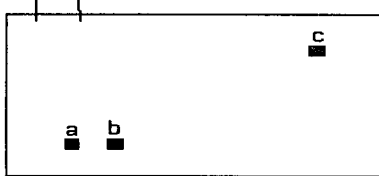


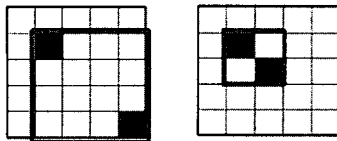
그림 3. Basic\_Group 알고리즘의 문제점

이를 해결하기 위해 본 논문에서는 그룹의 적절성을 판단하기 위하여 복잡성(C(g<sub>i</sub>): Complex Rank of group)이라는 개념을 도입하였다.

3.3 그룹의 복잡성

그룹의 복잡성을 판별하기 위하여 본 논문에서는 실내 디자인 분야 또는 영상 처리 분야에서 실내 구조의 복잡한 정도를 계산하는데 사용하는 히스토그램 방식을 사용한다[6].

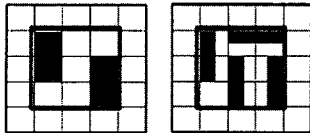
히스토그램은 영상의 전체 명암 값의 분포를 담을 수 있는 도구로써 이를 이용하면 영상의 명암도 분포 상태를 알 수 있다. 히스토그램 구성 방법은 0부터 255까지의 명암 값을 인덱스로 하고, 영상을 구성하고 있는 각 화소의 명암 값에 해당하는 개수를 빈도수로 간주하여 1씩 증가시키는 방법을 사용한다. 이때 0은 검은색, 255는 흰색을 나타낸다. 따라서 장애물을 검은색 픽셀로 표시하고 청소 대상 공간을 흰색 픽셀들의 집합으로 가정하여 처리하면 된다. 이때 각 색상에 대한 픽셀 값의 분포도를 측정하여 이미지의 복잡성을 판단할 수 있다. 히스토그램을 분석하면 평균값이 이미지의 경연색 픽셀이 많으면 낮아진다. 즉, 평균값이 낮을수록 복잡한 구조를 가진다고 볼 수 있다.



(a) 복잡성이 낮은 그룹 (b) 복잡성이 높은 그룹  
그림 4. 히스토그램에 의한 복잡성 판단의 적절한 예

그림 4와 같이 동일한 장애물이라도 그룹의 전체 면적이 다르기 때문에 히스토그램의 평균값이 달라 복잡성이 달라진다. 그렇기 때문에 히스토그램의 평균값이 그룹의 적절성을 판단하는 타당성 있는 기준이라 할 수 있다.

그러나 히스토그램 방법은 그림 5와 같이 구조는 다르나 면적이 같은 경우에 문제점이 발생한다. 그림 5를 보면 직관적으로 (b)가 (a)보다 복잡함을 알 수 있다. 그러나 히스토그램의 평균값을 보면 (b)와 (a)는 같다. 이것은 히스토그램이 지리적 위치나 모양을 전혀 고려하지 않기 때문이다.



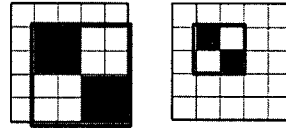
(a) 단순한 장애물 경우 (b) 복잡한 장애물 경우  
그림 5. 히스토그램에 의한 복잡성 판단의 부적절한 예

그러므로 그룹의 복잡성을 판단하기에는 히스토그램에 의한 색상 분포만으로는 최상의 결과를 기대하기 어렵다.

그래서 히스토그램과 함께 장애물의 구조적인 복잡성을 판단할 수 있는 꼭짓점의 수를 측정하여 장애물의 외양적 구조의 복잡성을 측정한다. 즉 꼭짓점의 수가 많으면 구조적으로 복잡한 공간이고 꼭짓점의 수가 적다면 단순한 구조를 가진다고 판단하는 것이

다. 동근모양의 장애물도 여러 픽셀이 찍혀있는 것이므로 매우 많은 꼭짓점의 수를 가지고 있다고 판단 할 수 있다.

하지만 히스토그램과 꼭짓점의 수를 고려하여도 그림 6과 같은 그룹에서는 (a), (b)그룹간의 청소 대상면적이 차이가 나지만 복잡성은 차이가 나지 않는다. 그래서 마지막으로 고려한 것은 전체 청소 대상 면적에 대한 그룹 내 청소 대상 면적의 비율이다. 만약 동일한 색상 분포를 가지고 꼭짓점의 수가 같은 그룹이 존재한다면 두 그룹 중 청소 대상 면적이 큰 그룹을 먼저 청소해야 한다.



(a) 청소대상면적이 큰 경우 (b) 청소대상면적이 작은 경우  
그림 6. 히스토그램과 꼭짓점의 비율에 의한 복잡성 판단의 부적절한 예

정리하면, 그룹의 복잡성은 히스토그램의 평균값, 그룹 내 장애물의 외양적 구조를 나타내는 꼭짓점의 수, 그리고 청소 대상면적의 크기를 조합하여 판단한다. 문제점은 이 세 가지 값들의 조합 방법이다. 이를 위해 본 논문에서는 각각의 값들을 최소 값을 0 최대 값을 1이 되는 정규화된 비율로 변환 한다. 그리고 이 세 값의 평균값을 복잡성으로 결정한다. 복잡성 값이 0 이면 장애물이 없는 상태를 말하는 것이고 1이라면 가장 복잡한 구조이다. 이를 수식으로 정리 하면 다음과 같다.

$$C(g_i) = \frac{(1 - \frac{Hmean(g_i)}{255}) + (\frac{Asum(g_i)}{Asum(G)}) + (1 - \frac{N(g_i)}{N(total)})}{3}$$

수식에 대하여 보면 Hmean(g<sub>i</sub>)는 특정 그룹(g<sub>i</sub>)의 히스토그램의 평균값을 나타낸다. 히스토그램은 0에서 255의 평균값을 산출하는데 이 값을 0에서 1사이의 비율로 변환하여야 한다. 주의하여야 할 점은 평균값의 255가 0으로 평균값의 0 이 1로 변환 되어야 한다는 것이다. 히스토그램 값이 127.5인 경우에는 변환 값이 0.5가 된다.

다음으로 장애물의 꼭짓점 수는 전체 장애물들의 꼭짓점 수의 합 Asum(G)에 대한 특정 그룹(g<sub>i</sub>)의 꼭짓점의 수의 합 Asum(g<sub>i</sub>)의 비율로 산정한다. 전체 꼭짓점수의 합과 특정 그룹 내의 꼭짓점의 수의 합이 같다면 1, 그룹 내에 장애물이 없다면 0으로 나타낸다. 마지막으로, 청소 대상 면적의 비율은 전체 청소 대상면적 N(total)에 대한 특정 그룹(g<sub>i</sub>)의 청소대상 면적 N(g<sub>i</sub>)의 비율로 측정한다. 만약 전체공간이 하나의 그룹으로 되었을 경우에는 0이 되고 청소 대상 면적이 작을수록 1에 가까워 질 것이다. 따라서 특정 그룹(g<sub>i</sub>)의 복잡성 C(g<sub>i</sub>)는 위에서 언급한 세 요소들의 평균값으로 결정 된다.

3.4 그룹 간 해제 및 통합

다음으로는 복잡성 C(g<sub>i</sub>)를 이용하여 전체 그룹들을 재조정하는 것이다. 이 작업은 크게 두 가지로 나누어진다. 먼저 어떤 그룹이 복잡성의 기준 k에 미치지 못한다면 이 그룹을 축소하거나 해제하는 과정(Release\_Group)과 두 번째는 그룹의 범위를 확장 조절(Merge\_Group) 하는 단계이다. 이 각 단계에 대하여 구체적으로 보면 그림 7과 같다.

Algorithm Release\_Group:

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

그림 7. Release\_Group 알고리즘

Release\_Group은 실내 공간의 전체 그룹에 대해 순차적으로 각 그룹별 복잡성을 계산하고, 복잡성이 k보다 작으면 그룹 내에서 가장 긴 링크를 해제하고 남은 장애물들로 그룹을 재형성 후 그룹의 복잡성을 다시 계산한다. 이런 과정을 복잡성이 k이상이 될 때까지 반복한다. 다음의 Merge\_Group 알고리즘은 그림 8과 같다.

Algorithm Merge\_Group:

- step 1. Select the Nearest Neighbour group of the group g<sub>i</sub> in G
- step 2. Make a virtual group v of g<sub>i</sub> and its nearest neighbour group
- step 3. Compute the complexity C(v) of group v
- step 4. If (C(v) >= 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  $G$ 
    
```

그림 8. Merge\_Group 알고리즘

Release\_Group을 거친 각 그룹들은 복잡성이 적어도  $k$  이상일 것이다. 이런 그룹들을 대상으로 인접한 그룹들을 병합한 가상 그룹의 복잡성을 계산하여  $k$  이상일 경우에 실제로 합병하게 된다. 이렇게 함으로써 소규모 그룹의 수를 줄여 소규모 그룹으로 인하여 생성되는 복잡한 구조를 최소화 한다.

### 3.5 청소 알고리즘

실내 공간의 장애물들에 대한 그룹 조절 작업이 끝나면 생성된 그룹을 기반으로 청소를 한다. 청소는 모든 그룹들의 복잡성이 계산되어 있으므로 이를 바탕으로 하여 먼저 그룹에 속하지 않은 전체 구역을 청소, 그 다음부터 복잡성이 낮은 그룹 순서대로 청소를 실시한다. 만약 복잡성이 같은 그룹이 있다면 현재 로봇의 위치에서 가장 가까운 위치에 있는 그룹을 먼저 청소하는 것이 바람직할 것이다. 이러한 과정을 정리한 것이 그림 9 이다.

#### Algorithm Group-k:

- step 1. Build a map for the space to be clean and Initialize the position of the cleaning robot
- step 2. Call the procedure *Basic\_Group*
- step 3. Call the procedure *Release\_Group*
- step 4. Call the procedure *Merge\_Group*
- step 5. Sort and make a list  $L$  for the all 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.

그림 9. Group-k 알고리즘

## 4. Group-k 알고리즘 성능 평가

### 4.1 시뮬레이터의 작성 및 환경 설정

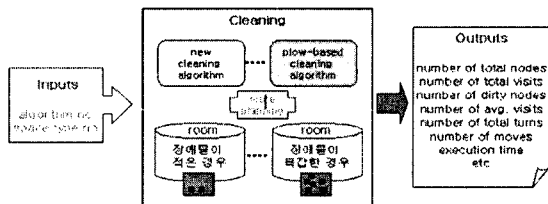


그림 10. 청소 알고리즘 성능 분석용 시뮬레이터 구성

시뮬레이션은 레드햇 리눅스 8.0을 탑재한 펜티엄Ⅳ 시스템에서 이루어졌다. 시뮬레이터는 C언어를 이용하여 직접 작성하였으며, 그림 10에서 보는 바와 같이 청소 알고리즘 모듈, 맵 모듈, 입출력 모듈 등으로 구성 된다. 복잡성의 기준인  $k$ 값은 0.5로 초기화 하였다. 로봇의 현재 위치는 @으로 표현 하였다. 로봇의 초기 위치는 좌표값(1,1)으로 설정하였다. 그리고 한 좌표는 실제공간에서 약 20cm를 나타내도록 설계 하였으며, 그림 11에서와 같이 맵에서 #은 장애물 또는 벽 0은 청소하여야 하는 공간이다. 이외에도 알고리즘 별로 성능을 측정하기 위해 여러 가지 정보를 출력한다.

### 4.2 실험 결과 분석

그림 12는 장애물이 복잡한 공간에서 바둑판식과 Group-k 알고리즘의 실험결과이다. 그래프는 시간의 흐름에 따른 청소 완료된 면적 비율이다. Group-k 알고리즘은 청소종료시간은 더 오래 걸리나, 단 기간에 많은 부분을 청소할 수 있음을 보여준다. 이처럼 Group-k 알고리즘은 복잡한 공간에서 넓은 부분을 바둑판식 알고리즘보다 빠르게 청소하고, 장애물이 없는 공간에서는 바둑판식과 동일한 성능을 나타내게 되어 실내 공간의 복잡도와 상관없이 다양한 공간에서 적용될 수 있는 실용적 알고리즘임을 알 수 있다.

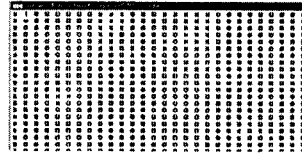


그림 11. 청소 알고리즘 분석을 위한 대상 실내 구조 맵 구현 예

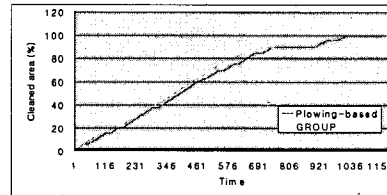


그림 12. 청소 완료율 그래프

## 5. 결론 및 향후 연구

본 논문에서는 바둑판식 청소 알고리즘의 문제점을 해결하는 새로운 알고리즘인 Group-k 알고리즘을 제안하였다. Group-k 알고리즘은 장애물들을 그룹으로 형성하여 복잡성이 낮은 공간부터 먼저 청소하도록 하여 바둑판식이 가지고 있던 문제점을 해결 하였다. 시뮬레이션을 통해 성능을 평가한 결과 복잡성이 높은 곳일수록 Group-k 알고리즘은 바둑판식 알고리즘보다 좋은 성능을 나타내었으며, 장애물이 없는 공간에서는 바둑판식 알고리즘과 대등한 성능을 나타내어 실용적인 알고리즘임을 보여 주었다.

특정 그룹의 복잡성을 나타내기 위하여 적합한 히스토그램, 꼭짓점의 수, 청소대상면적 등의 세 요소들 중에서 상대적으로 그룹을 형성 하는데 민감한 요소는 어느 것인지에 대한 연구가 필요하겠고,  $k$ 값의 결정에 어떤 영향을 미치는지에 대한 관찰이 추가적으로 진행될 것이다.

## 6. 참고문헌

- [1] Gaurav S. Sukhatme, Maja J. Mataric "Embedding Robots Into the Internet", *Communications of the ACM*, May 2000/Vol. 43, No 5
- [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] Sylvia C. Wong, Bruce A. MacDonald "A topological coverage algorithm for mobile robots", In *Proceedings of the 2003 IEEE/RSI Int. Conference on Intelligent Robots and Systems*, Las Vegas, Nevada, October 2003
- [6] 윤상훈, 박세훈, 최병준, 이연정, "Path Planning for Cleaning Robots: A Graph Model Approach", In *Proceedings of the International Conference on Control, Automation and Systems*, pp.2199-2202, October 17-21, Cheju National Univ. Jeju, Korea, 2001
- [7] Sven Koenig, Yaxin Liu "Terrain Coverage with Ant Robots: A Simulation Study", In *Proceedings of the ACM AGENTS'01*, May 28-June 1, 2001, Montreal, Quebec, Canada
- [8] Andrew Howard, Sajid Siddiqi, Gaurav S. Sukhatme "An Experimental Study of Localization Using Wireless Ethernet", In *Proceeding of the 4th International conference on Field and Service Robotics*, July 14-16, 2003
- [9] Maxim A. Batalin and Gaurav S. Sukhatme "Coverage, Exploration and Deployment by a Mobile Robot and Communication Network", In *Proceedings of the 2nd International Workshop on Information Processing in Sensor Networks, Palo Alto Research Center(PARC)*, Palo Alto, April 22-23, 2003
- [10] 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, 2002년 4월