

영상 감시 시스템을 위한 에이전트 기반의 자동화된 카메라 배치☆

Agent-based Automatic Camera Placement for Video Surveillance Systems

변 유 인* 남 윤 영** 조 위 덕***
Uin Burn Yunyoung Nam We-Duke Cho

요 약

본 논문은 에이전트 기반의 시뮬레이션을 이용하여 최적의 카메라 배치 방법을 제안한다. 공간 중요도를 분석하고 공간을 효율적으로 커버하기 위해, 공간분류와 이동하는 사람의 패턴 분석기반으로 에이전트 기반의 시뮬레이션을 수행하였다. 에이전트 기반의 카메라 배치 기법은 카메라 성능뿐만 아니라 경로탐색 알고리즘으로부터 추출한 공간 중요도를 고려하여 개발하였다. 이 기법은 설치비용을 고려하여 최적의 카메라 수를 결정할 뿐만 아니라 카메라의 위치와 방향을 조정한다는 것을 보인다. 제안한 방법을 검증하기 위해 시뮬레이션 결과를 실제 영상과 비교하였으며, 특정 공간에서 시뮬레이션한 실험 결과를 보인다.

ABSTRACT

In this paper, we propose an optimal camera placement using agent-based simulation. To derive importance of space and to cover the space efficiently, we accomplished an agent-based simulation based on classification of space and pattern analysis of moving people. We developed an agent-based camera placement method considering camera performance as well as space priority extracted from path finding algorithms. We demonstrate that the method not only determinates the optimal number of cameras, but also coordinates the position and orientation of the cameras with considering the installation costs. To validate the method, we compare simulation results with videos of real materials and show experimental results simulated in a specific space.

☞ KeyWords : surveillance system, camera placement, software agents, simulation, optimal camera placement

1. 서 론

유비쿼터스(Ubiquitous)[1]는 1993년 마크 와이저(Mark Weiser)에 의해 발표된 이후에 많은 형태의 기술과 문화적 키워드로써 모습을 달리하여

소개되었으며, 정보화시대 이후 급속도로 성장한 IT기술로 시공간의 제약을 초월하여 사용자에게 네트워킹 서비스를 지원해줄 수 있는 유비쿼터스 시대가 도래하였다. 유비쿼터스 시스템은 유비쿼터스 기술을 통해 어떤 목적을 성취하기 위한 사람, 객체, 그리고 프로세스 집합이다. 이러한 유비쿼터스 시스템을 개발하고, 구축하는 데에 있어서 비용, 규모, 복잡도 등 다양한 측면에서 어려움이 존재한다. 즉, 유비쿼터스 시스템의 구축에 필요한 H/W와 S/W의 비용뿐만 아니라 설계 오류를 수정하기 위한 비용이 발생하며, 유비쿼터스 시스템을 이루는 각 구성요소들 간 상호 작용이 복잡하기 때문에 시스템 구축에 수반되는 시행착오에 따른 시간적 비용적 문제와 개발자 경험에 의존

* 정 회 원 : (주)세나테크놀로지 주임연구원
byunja@gmail.com

** 정 회 원 : 아주대학교 유비쿼터스컨버전스연구소
연구교수 young022@gmail.com(교신저자)

*** 정 회 원 : 아주대학교 전자공학부 교수
wdukecho@gmail.com

[2009/03/18 투고 - 2009/04/01 심사(2009/07/08 2차)
- 2009/08/07 심사완료]

☆ 본 연구는 지식경제 프론티어 기술 개발사업의 일환으로 추진되고 있는 지식경제부의 유비쿼터스 컴퓨팅 및 네트워크 원천기반기술 개발사업의 10C2-T3-10M 과제로 지원된 것임

하는 문제가 발생하게 된다.

지능형 영상감시 시스템은 카메라들을 통해 사람과 객체들을 모니터링하고 위험상황에 대해 즉각적으로 대응하는 대표적인 유비쿼터스 시스템으로, 시뮬레이션을 통하여 시스템 엔지니어로 하여금 효과적인 유비쿼터스 시스템을 구축할 수 있는 방법에 대한 연구가 선행되어야 한다. 지금까지 지능형 영상 감시 시스템은 주로 효과적인 객체 인식이나 다중 카메라에서의 객체 추적 알고리즘에 대한 연구가 주였지만, 앞으로의 지능형 영상감시 시스템은 단일 카메라의 기능 개발이 아닌 지능공간에서의 다수의 카메라가 유기적으로 협업된 형태의 시스템으로 제한된 비용에서 효과적으로 객체 인식 및 추적이 이루어질 수 있도록 최적의 카메라 배치에 대한 연구가 필요하다. 본 논문은 효율적인 카메라 배치를 위해 카메라의 FOV(Field Of View)를 기반으로 감시 영역의 커버리지를 계산하였으며, 가상 에이전트들의 이동패턴을 모델링 하여 공간중요도(Space Priority)를 계산하였다. 또한, 최소의 비용으로 공간을 커버할 수 있는 카메라 배치 알고리즘을 구현하고 실제 건물을 모델링하고 가상의 카메라들의 성능과 비용을 입력하여 시뮬레이션 하였다.

본 논문은 다음과 같이 구성된다. 2장에서는 관련 연구를 소개하고, 3장에서는 카메라 배치에 필요한 공간구성, 에이전트에 대한 모델링, 경로 탐색 알고리즘을 설명한다. 4장에서는 설정된 테스트베드를 모델로 수행한 시뮬레이션 과정과 도출된 카메라의 최적 배치 위치에 대해 기술한다. 5장에서는 실험 결과를 보이고, 마지막 6장에서 결론을 맺는다.

2. 관련연구

지금까지 많은 연구자들에 의해 실제 공간을 가상의 3차원 공간으로 설계하고 시뮬레이션을 하기 위한 여러 가지 방법들이 연구되었다. UBIWISE [2]와 TATUS [3]는 사용자가 공간을 직

접 디자인하고 u-Device를 정의하여 배치와 테스트를 하기 위한 틀이다. 기존의 시뮬레이터들은 공간과 디바이스 디자인에 대한 지원에 초점이 맞춰진 틀로써, 환경적인 공간상태, 지형재질, 컨텍스트(Context) 등 물리적 조건의 분석이 선행되지 않았다는 문제점이 있었다.

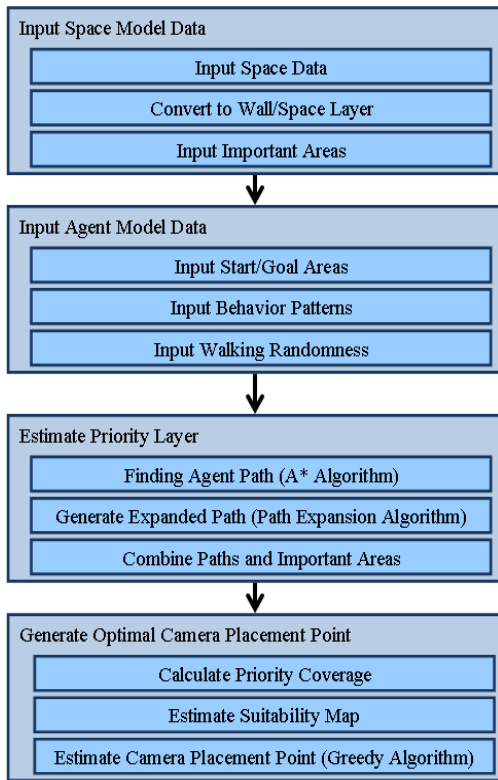
이를 보완하여 일본 시가대학과 나라첨단기술연구대학에서는 UbiReal [4]이라는 유비쿼터스 지능공간 시뮬레이터를 발표하였다. UbiReal은 가상의 유비쿼터스 지능공간을 시뮬레이션하는 어플리케이션으로써, 현실의 디바이스와 가상의 디바이스간의 연관성, 3차원 공간과 디바이스의 상태 모니터링, 벽, 장애물에 대한 무선 통신의 장애, 온도변화, 환기 등의 변화를 고려하였다.

Miquel [5]은 공간의 지형과 공간 내에 존재하는 오브젝트들에 대한 무선센서와 이동성을 고려하여, 광역 공간의 시뮬레이터를 연구하였다. Ram [6]은 다양한 멀티 센서를 감시시스템에 적용하기 위한 방법을 연구하였다. 서비스 적용 범위와 서비스 효율이 다른 초음파, 음차 센서 등을 감시시스템에 적용하기 위해 기존의 감시시스템에 사용된 센서들에 대한 배치방식과의 차이에 대해 서술하였다.

Horster [7]는 다중 영상센서에 대한 최적의 배치 방법에 대한 연구를 하였다. 영상센서의 해상도(Resolution)에 따라 카메라의 최대 커버리지(coverage)를 위하여, 카메라 위치에 대해 그리디 알고리즘(Greedy Algorithm)과 듀얼 샘플링 알고리즘(Dual Sampling Algorithm)을 적용하였다. Zhao [8]는 멀티카메라의 FOVs를 측정하여 태그를 통해 카메라들의 배치에 대한 효율성을 분석하는 연구를 하였다. 그러나, 기존의 연구들은 공간의 구조와 영역만을 고려하고 사람의 이동패턴과 공간의 영역별 중요도를 고려하지 않았기 때문에 최적의 카메라의 배치 방법이라고 보기 어렵다.

3. 공간과 에이전트 모델링

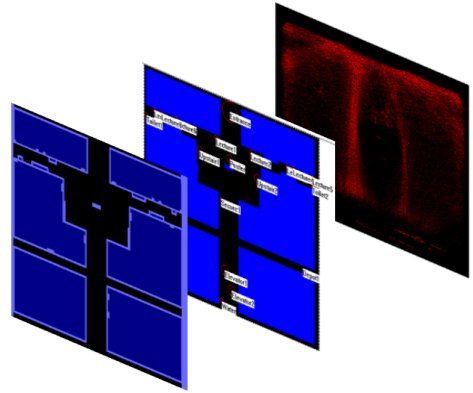
카메라의 위치와 방향에 대한 최적의 배치를 계산하기 위해, 우선 공간을 모델링하고 사람들이 이동하는 경로를 추출한 후, 사람들의 유동량을 계산해야 한다. 그림 1은 이러한 카메라 배치를 위한 시뮬레이션 순서이다. 우선, 모델링된 지형정보와 에이전트 데이터를 입력하고, 공간중요도를 계산하여, 최적의 카메라 배치위치를 판단한다.



(그림 1) 카메라 배치를 위한 시뮬레이션 순서

3.1 공간 모델링

본 논문에서는 시뮬레이션에 사용하는 공간을 2차원 맵으로 모델링 하였으며, 보행자의 이동 경로와 유동량을 계산하기 위해 가상의 에이전트를 모델링 하였다. 공간은 그림 2와 같이 3 가지 레이어로 구성하였다.



(그림 2) 공간을 구성하는 레이어 (왼쪽부터 지형 레이어, 영역 레이어, 공간중요도 레이어)

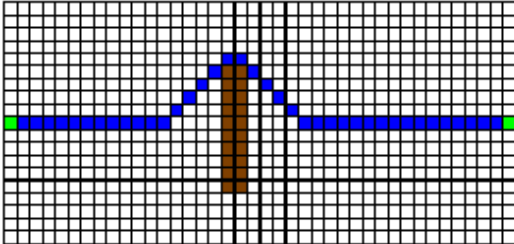
지형 레이어는 공간에서 사람이 이동 가능할 영역과 이동이 불가능 영역으로 구분하였으며, 이진 배열 형태로써 경로탐색 알고리즘에 적용하였다. 영역 레이어는 특정 영역을 원의 형태로 구성하여 속성정보와 함께 저장된 레이어이다. 영역 레이어의 속성은 위치, 범위, 유동량, 이동확률로 구성된다. 유동량은 영역을 통과하는 사람의 이동량이며, 이동확률은 한 영역에서 다른 영역으로 이동할 확률이다. 공간중요도 레이어는 지형 레이어와 영역 레이어로부터 생성되고, 경로탐색으로 계산된 공간중요도를 통해 업데이트된다.

3.2 에이전트 모델링

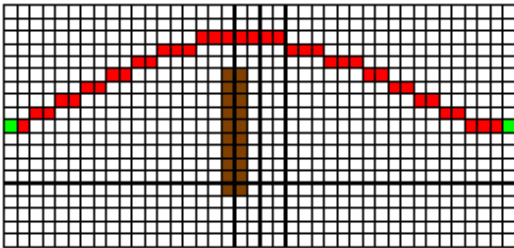
3.2.1 에이전트의 경로 탐색 알고리즘

에이전트는 공간에서 이동 가능한 모든 경로를 생성하며, 이를 통해 이동 가능한 영역과 빈번도가 높은 이동 경로를 추출한다. 한 지점에서 다른 지점으로 이동하더라도 사람의 감성, 사전지식, 주변 환경 등 다양한 요인에 따라 경로선택의 우선순위가 달라지기 때문에 여러 가지 경로가 나타난다. 그림 3에서 보는 바와 같이, 최단거리를 우선순위로 하는 경로탐색 알고리즘으로 구한 경로는 사람의 경로선택과 다르게 나타나는데, 이것

은 에이전트가 모든 지형을 학습하고, 시야와 방향 전환 등을 고려하지 않았기 때문이다.



(가) A* 알고리즘을 이용한 이동경로



(나) 사람의 이동경로

(그림 3) 경로탐색 알고리즘을 이용하여 추출한 이동경로와 사람의 이동 경로

본 논문에서는 이러한 에이전트의 이동경로와 사람의 이동경로에 대한 차이를 줄이기 위하여, 경로탐색 알고리즘으로 추출한 경로에서 주변의 가용 경로를 예측하여 다양한 경로들을 생성하였다. 에이전트는 알고리즘 1을 이용하여 모든 목적지에서 이동 가능한 모든 영역을 검사하는 방법으로 경로탐색 시뮬레이션을 수행한다.

<알고리즘 1> 에이전트 경로탐색 알고리즘

1. 선택되지 않은 목적지가 존재하지 않는다면 시뮬레이션을 끝내고, 그렇지 않다면 현재위치에서의 확률분포에 근거하여 목적지를 선택한다.
2. 목적지까지의 경로를 경로탐색 알고리즘을 통해 탐색한다.
3. 탐색된 알고리즘상의 경로에서 영역 속성정보

에 따라 경로확장 알고리즘을 사용하여 경로를 확장한다.

4. 구한 모든 확장 경로 상에 존재하는 이동 가능한 지역의 공간중요도를 영역 확률분포에 따라 공간중요도 레이어에 업데이트한다.
5. 목적지를 현재위치로 변경하고, 1번으로 돌아간다.

3.2.2 에이전트의 경로 확장 알고리즘

A* 알고리즘[9]은 휴리스틱 탐색(Heuristic Search)[10]과 그래프 탐색(Graph Search) 알고리즘을 기반으로 최단경로와 최소비용으로 경로를 탐색한다. 본 연구에서 시뮬레이션에 적용한 A* 알고리즘은 다음과 같다.

<알고리즘 2> A* 알고리즘

1. 시작노드 A를 리스트 OPEN에 넣는다. A를 최초의 탐색트리 T에 저장한다.
2. 빈 리스트 CLOSED를 정의한다.
3. OPEN이 비어 있는지 확인해보고 비어있다면 탐색 실패를 반환한다.
4. OPEN에서 첫 번째 노드를 선택하여 An으로 정하고 CLOSED에 넣는다.
5. An이 목표 노드이면 A에서 An까지의 경로를 추적하여 반환한다.
6. 노드 An에 연결된 모든 노드들을 탐색 트리 T에 저장한다.
7. 연결된 모든 노드들을 OPEN에 넣는다.
8. 휴리스틱과 기타 요소값에 따라 OPEN 내용을 재정렬한다.
9. 단계 3으로 되돌아간다.

본 논문에서는 A* 알고리즘을 이용하여 2차원 그리드 기반 맵의 방향을 8 가지로 구분하고 휴리스틱 평가함수를 다음과 같이 정의함으로써, 에이전트 경로탐색에 A* 알고리즘을 사용하였다.

$$f = g + h \quad (1)$$

g 는 시작 노드에서 현재 노드까지의 이동 비용의 합이며 h 는 현재 노드에서 도착 노드까지의 장애물을 제외한 이동 비용의 합이다. 따라서 g 와 h 의 합인 f 가 A* 알고리즘에서 경로탐색에서 우선순위를 정하는 최종 기준이 된다. 이때 사용한 이동 비용은 상하 좌우 방향은 1, 대각선 방향은 1.4의 비용으로 정하였으며 노드간의 이동 비용은 맨하탄 기법(Manhattan Method)[11]으로 계산하였다.

A* 알고리즘을 통해 생성한 이동 경로와 사람의 이동 경로는 차이가 발생하기 때문에 이러한 차이를 줄이기 위해, A* 알고리즘으로 추출한 경로 주변의 이동 가능한 공간 중 사람이 이동할 수 있는 영역을 추론하여 공간중요도 레이어에 중요도를 재계산하였다. 경로 확장 알고리즘의 추론 과정은 다음과 같다.

<알고리즘 3> 에이전트 경로 확장 알고리즘

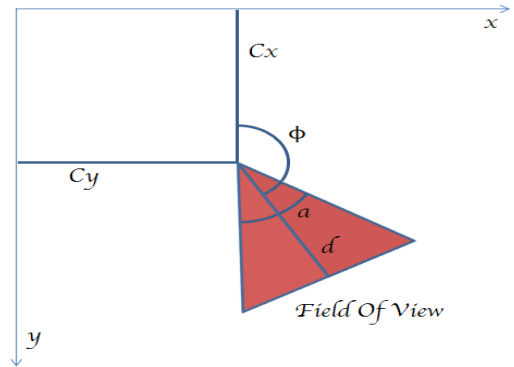
1. 영역 속성정보에 따른 출발지점, 도착지점의 영역범위 $r1, r2$ 을 구한다.
2. 출발지점의 영역범위 $r1$ 의 크기에 따라 확장 출발지점개수 n 을 구하고, 영역 범위 내에 n 개의 확장 출발지점 위치를 정한다.
3. 도착지점의 영역범위 $r2$ 의 크기에 따라 확장 출발지점개수 m 을 구하고, 영역 범위 내에 m 개의 확장 출발지점 위치를 정한다.
4. 출발지점을 포함한 확장 출발지점 $n+1$ 개, 도착지점을 포함한 확장 도착지점 $m+1$ 개에 대해 A* 알고리즘을 적용하여 $(n+1)*(m+1)$ 개의 추가 경로를 탐색한다.
5. 각 경로들의 선에 의해 둘러 쌓인 그리드들을 검색하여 확장영역으로 추가한다.
6. 추가된 모든 경로에 대한 Grid와 5번에 구한 그리드들을 확장영역으로 리턴한다.

4. 카메라 배치

지금까지, 지형정보를 모델링하고 가상의 에이전트들을 통해 경로를 추출하고 공간중요도를 계산하였다. 이러한 값으로 FOV가 커버하는 경로 비율과 경로확률을 계산하여 카메라 배치를 결정한다.

4.1 카메라의 FOV 모델링

시뮬레이터에서 사용할 카메라의 FOV는 그림 4에서 보는 바와 같이, 카메라의 위치를 기준으로 인식거리와 가용각도로 이루어진 삼각형의 구조를 가진다. 카메라의 위치는 2차원 그리드 맵에서 (x, y) 의 정수좌표이고 $0^\circ \sim 360^\circ$ 의 각도를 갖는다. 즉, 카메라의 FOV는 카메라의 위치 (C_x, C_y) , 위치 각도 Φ , 인식거리 d , FOV 각도 a 의 구성요소를 갖는다.



(그림 4) 카메라 FOV 모델링

카메라 FOV를 구하기 위하여 식 2와 같이 카메라 위치를 원점으로 이동시킨 새 좌표에 대입시킨다.

$$x' = x - C_x, y' = y - C_y \quad (2)$$

x 축을 기준으로 대칭된 삼각형으로 변환하게

되면 다음과 같은 식을 얻을 수 있다.

$$x'' = \cos(\Phi) \cdot x' + \sin(\Phi) \cdot y' \quad (3)$$

$$y'' = -\sin(\Phi) \cdot x' + \cos(\Phi) \cdot y' \quad (4)$$

따라서, 위의 식으로부터 아래와 같은 식을 계산할 수 있다

$$l1: x'' \leq d \quad (5)$$

$$l2: y'' \leq \frac{a}{2d} \cdot x'' \quad (6)$$

$$l3: y'' \geq -\frac{a}{2d} \cdot x'' \quad (7)$$

(5)(6)(7)의 식은 카메라의 인식거리와 FOV 각도에 대한 식으로 (2)(3)(4)의 값들을 대입하면 아래 (8)(9)(10)과 같이 변환할 수 있으며, 시물레이션에서는 아래의 식과 일치하는 영역을 카메라의 시야한계로 정한다.

$$\cos(\Phi) \cdot (x - C_x) + \sin(\Phi) \cdot (y - C_y) \leq d \quad (8)$$

$$-\sin(\Phi) \cdot (x - C_x) + \cos(\Phi) \cdot (y - C_y) \leq \frac{a}{2d} \cdot [\cos(\Phi)(x - C_x) + \sin(\Phi) \cdot (y - C_y)] \quad (9)$$

$$-\sin(\Phi) \cdot (x - C_x) + \cos(\Phi) \cdot (y - C_y) \leq -\frac{a}{2d} \cdot [\cos(\Phi)(x - C_x) + \sin(\Phi) \cdot (y - C_y)] \quad (10)$$

FOV를 그리드에 기반한 영역 소거 방법으로 모델링 하여 실제 FOV 내의 오브젝트 인식률을 추정하거나 장애물에 대한 시야간섭을 고려할 수 없다. 따라서, 오브젝트에 대한 인식률 계산을 위해, FOV 영역 내에 장애물이나 벽과 같은 보행불가 영역은 FOV 내에서 차지하는 비율에 따라 중

요도를 다르게 할당함으로써 시야간섭을 고려하였다. FOV 내의 영역에 대한 이동 가능한 영역의 공간 중요도는 식 11과 같다.

$$priority_{WA}(x, y) = \frac{WalkableGridRatio_{FOV}}{WalkableGridRatio_{SpaceLayer}} \quad (11)$$

*WalkableGridRatio*는 보행가능공간의 비율로써 전체 지형 레이어 상에서 FOV내에 있는 비율을 계산하여 공간중요도를 재계산한다. A* 알고리즘으로 추출한 경로에 해당하는 그리드는 FOV 내에 해당 경로를 포함하는 정도에 따라 공간 중요도가 결정된다. 영역 레이어의 속성정보에 저장된 해당 경로의 확률에 따라 경로의 중요도를 계산하여 경로확률(Path Probability)을 정하고, 경로 상에 나타나는 모든 그리드의 수를 경로비율(Path Ratio)라고 할 때, 지형 레이어 전체의 경로비율에서 FOV가 커버하는 경로비율과 경로확률로 공간 중요도를 결정한다. A* 알고리즘으로 추출한 경로의 공간 중요도 계산식은 식 12와 같다.

$$priority_{A^*}(x, y) = \frac{PathRatio_{FOV}}{PathRatio_{SpaceLayer}} \times Pathprobability \quad (12)$$

확장경로에 대해서는 A* 알고리즘으로 추출한 경로와 같은 방법으로 계산하고 파생된 확장경로의 수에 따라 경로확률을 나누어 계산하며 계산식은 식 13과 같다.

$$priority_{EP}(x, y) = \frac{PathRatio_{FOV}}{PathRatio_{SpaceLayer}} \times \frac{Pathprobability}{ExtendPaths} \quad (13)$$

벽과 장애물의 경우 공간 구분으로 장애물의 그리드 하나가 FOV를 얼마나 차지하는지를 파악

하는 것은 어렵기 때문에, 벽과 장애물의 경우는 FOV 내의 이동 불가능한 영역인 이동불가지형(Unwalkable Grid)의 비율을 계산하여 공간중요도를 감소시켰다. 계산식은 식 14와 같다.

$$priority_{UWA}(x, y) = \frac{UnwalkableGridRatio_{FOV}}{UnwalkableGridRatio_{SpaceLayer}} \quad (14)$$

4.2 배치 위치 선택

카메라의 배치 위치를 선택하기 위해 그리디 알고리즘을 이용하였다. 시뮬레이션 공간 모델의 영역 속성에서 공간중요도와 에이전트의 경로 탐색 알고리즘으로부터 얻은 경로정보를 바탕으로 공간중요도 레이어를 추출한다. 카메라의 배치에 대한 위치 선택은 공간중요도 레이어를 통해 배치 위치를 선택하게 되며, 공간 모델의 모든 좌표 점들을 비교하여 카메라 FOV를 통해 공간 중요도가 가장 높은 위치를 정한다. 최적의 위치는 각 좌표 점을 기준으로 적합도(Suitability)를 계산하여 찾아내며, 적합도는 카메라의 비용과 성능에 따라 다르기 때문에, 모든 좌표 점과 카메라 타입에 따른 적합도 계산식은 아래 식 15와 같다.

$$Suitability(x, y) = \frac{priority_{inFOV}ofcamera_n}{CostofCamera_n} \quad (15)$$

적합도는 해당 그리드 좌표에서 모든 방향에 대해 FOV 내에 공간중요도가 최대인 위치 각도를 계산한 후, 그 위치 각도에서 얻은 공간중요도를 카메라 비용으로 나누어 계산한다. 가장 높은 적합도를 갖는 그리드의 순서대로 카메라 위치가 결정되고, 적합도 맵은 재구성 된다.

공간중요도 레이어에 설치된 카메라들의 FOV를 통해 공간 모두가 커버된다면 카메라 배치는 종료되지만, 공간중요도를 충족하는 것만으로 최적의 카메라 배치라고 할 수 없기 때문에 본 논문

에서는 카메라의 최적 배치를 위해 배치 한계비용을 고려하였다. 배치 한계비용은 한계비용을 초과하지 않는 최대의 카메라 개수이며, 알고리즘은 아래와 같다.

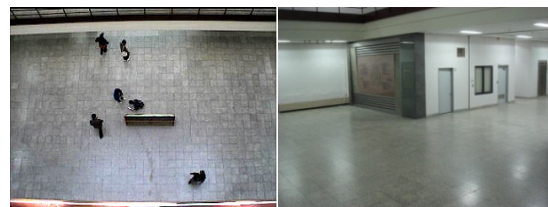
<알고리즘 4> 카메라 배치 알고리즘

1. 그리디 알고리즘에 의한 모든 공간 맵에서 카메라 FOV에 의한 커버리지와 비용을 고려한 적합도 맵을 구한다.
2. 적합도 맵에서 가장 높은 공간 중요도의 그리드 좌표를 선택한다.
3. 카메라를 배치하고 배치 리스트에 업데이트한다.
4. 배치개수가 임계값 이상이면 배치를 종료한다.
5. 배치된 카메라 위치 주변의 적합도 맵을 초기화 한다.
6. 2번으로 돌아간다.

5. 시뮬레이션

5.1 시뮬레이션 환경

카메라의 최적화된 배치를 위해 건물의 1층 내부를 모델링하고 시뮬레이션 하였다. 공간내의 이동하는 사람들의 수와 이동경로를 추출하기 위해서, 4층에 카메라를 설치하였다. 설치된 카메라를 통해 24 시간 동안 모니터링을 하여 시뮬레이션을 위한 확률모델로 사용하였다.



(가)

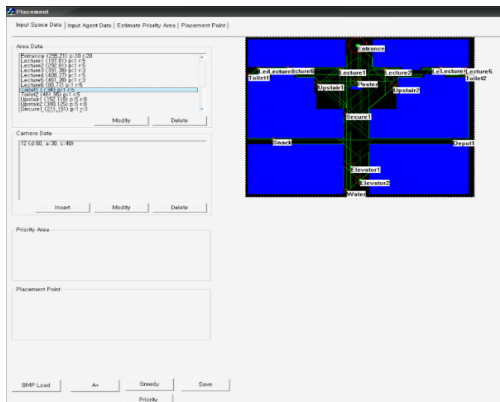
(나)

(가) 위에서 아래로 내려보는 카메라 뷰

(나) 옆에서 본 카메라 뷰

(그림 5) 실험공간

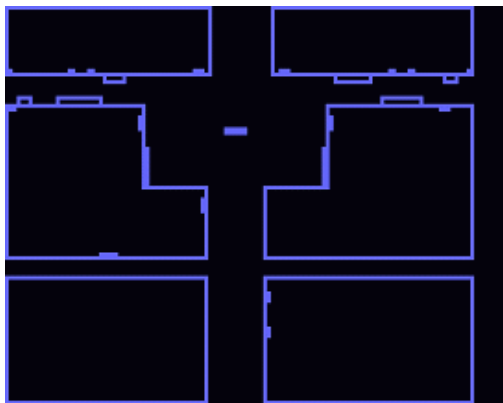
시뮬레이션 툴은 Visual C++ 6.0로 개발하였으며, CPU 2GHz와 1G RAM의 PC 2대를 이용하여 시뮬레이션 하였다. 그림 6에서 보는 바와 같이 영역과 공간데이터 입력을 위한 인터페이스를 구현하고, 경로탐색과 배치위치선택 알고리즘을 적용하여 시뮬레이션하였으며, 시뮬레이션을 완료하는데 약 6시간 소요되었다.



(그림 6) 시뮬레이션 툴

5.2 시뮬레이션 데이터

시뮬레이션에 사용된 데이터는 건물의 2차원 도면을 이용하여 모델링하였으며, 지형 레이어로 전달하는 과정을 거쳐 시뮬레이션하였다. 모델링된 도면은 그림 7과 같다.



(그림 7) 모델링된 도면

모니터링을 통해 얻은 데이터를 바탕으로 공간의 각 영역을 설정하고 에이전트들의 이동 목적지를 정하여 공간중요도를 위한 영역 레이어의 속성정보를 입력하였다. 입력한 건물의 20개의 영역은 표 1과 같다.

(표 1) 영역 정보

구분	X	Y	범위	유동량	비고
1	235	21	20	100	출입구
2	197	81	5	10	강의실1
3	292	81	5	10	강의실2
4	391	78	5	10	강의실3
5	408	77	5	10	강의실4
6	461	78	5	10	강의실5
7	88	77	5	10	강의실6
8	7	94	5	30	화장실1
9	461	95	5	30	화장실2
10	152	118	10	50	계단1
11	310	125	10	50	계단2
12	211	191	5	5	경비실1
13	28	78	5	10	강의실7
14	48	78	5	10	강의실8
15	435	257	5	5	연구실1
16	213	384	5	5	음수대
17	58	255	5	30	매점
18	235	110	8	10	벽보
19	221	323	10	50	엘레베이터1
20	240	356	10	50	엘레베이터2

건물의 구조에 따라 좌표와 범위를 정하였으며, 유동량에 따라 가중치를 정하였다. 영역 간의 이동 확률은 표 2와 같이 모니터링한 사람의 이동 횟수를 백분율로 변환하여 영역 속성정보에 적용하였다.

5.3 공간중요도

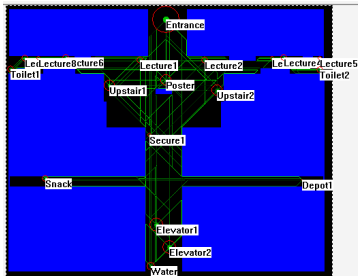
공간중요도 계산을 위해 표 1과 표 2의 데이터를 경로탐색 알고리즘에 사용하였다. 우선, 각 목적지간의 최소경로를 탐색하여 경로를 배열화하고, 경로확장탐색 알고리즘을 적용하여 기존 경로에서 확장 가능성이 있는 영역을 함께 고려하여 공간중요도를 계산하였다. 경로탐색 알고리즘의 적용결과는 그림 8과 같다.

(표 2) 이동 확률표

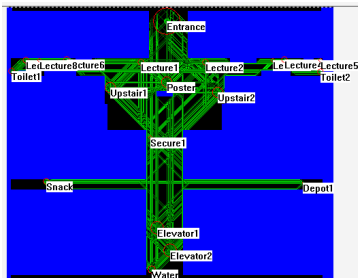
출발 도착	a	b	c	d	e	f	g	h	i	j	l	k	m	n	o	p	q	r	s	t
a	0	32	32	32	30	31	34	8	8	34	33	10	32	32	10	20	19	11	42	42
b	3	0	1	1	1	1	1	7	7	5	5	5	1	1	5	2	3	3	2	2
c	4	1	0	1	1	1	1	6	6	6	6	5	1	1	5	2	4	6	1	1
d	5	1	1	0	1	1	1	7	7	5	7	5	1	1	5	2	3	3	1	1
e	4	1	1	1	0	1	1	4	4	6	5	5	1	1	5	2	4	8	1	1
f	4	1	1	1	1	0	1	3	3	5	5	5	1	1	5	2	3	6	2	2
g	3	1	1	1	1	1	0	4	4	7	7	5	0	0	5	2	8	3	1	1
h	8	4	4	4	8	6	3	0	7	9	9	5	4	4	5	2	16	8	11	11
i	3	1	1	1	1	1	1	7	0	4	4	5	1	1	5	2	5	7	4	4
j	7	4	4	4	7	3	5	4	4	0	1	5	4	4	5	2	1	12	1	1
k	8	4	4	4	6	7	8	4	4	1	0	5	4	4	5	2	1	13	1	1
l	1	1	1	1	1	1	1	3	3	2	2	0	1	1	5	2	1	1	1	1
m	1	1	1	1	1	1	1	4	4	5	5	5	1	1	5	2	1	1	1	1
n	1	1	1	1	1	1	1	3	3	5	5	5	1	1	5	2	1	1	1	1
o	1	1	1	1	1	1	1	6	6	1	1	5	1	1	0	2	1	1	1	1
p	1	2	2	2	2	3	2	3	3	1	1	5	2	2	5	0	1	1	1	1
q	6	12	12	12	10	12	11	5	5	1	1	5	12	12	5	2	0	7	26	26
r	14	20	20	20	15	10	11	4	4	1	1	5	20	20	5	10	4	0	1	1
s	13	6	6	6	6	9	8	9	9	1	1	5	6	6	5	20	13	4	0	1
t	13	6	6	6	6	9	8	9	9	1	1	5	6	6	5	20	11	4	1	0

a:Entrance b:Lecture1 c:Lecture2 d:Lecture3 e:Lecture4 f:Lecture5 g:Lecture6 h:Toilet1 I:Toilet2 j:Upstair1
k:Upstair2 l:Secure1 m:Lecture7 n:Lecture8 o:Depot1 p:Water q:Snack r:Poster s:Elevator1 t:Elevator2

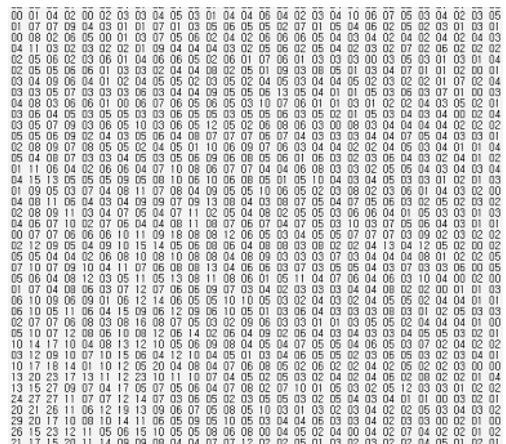
마지막으로 영역 레이어의 속성정보와 공간중요도 레이어를 이용하여 공간중요도를 배열의 형태로 계산하였다. 그림 9는 공간중요도의 일부를 보이고 있다. 공간 중요도 배열은 카메라 배치 위치를 결정하기 위한 입력 값으로 사용된다.



(가) A* 경로탐색 알고리즘을 적용한 영역 레이어



(나) 확장 경로탐색 알고리즘을 적용한 영역 레이어



(그림 9) 공간중요도 배열의 일부

(그림 8) 경로탐색 알고리즘 적용한 영역 레이어

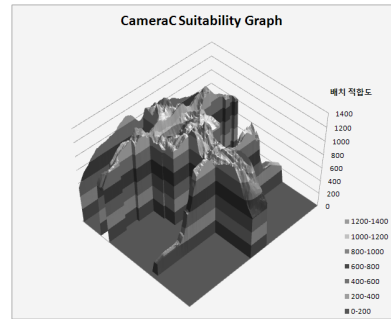
5.4. 카메라 배치위치 결정

카메라의 성능에 따라 카메라 배치가 달라질 수 있기 때문에 카메라의 종류를 세 가지로 나누어 시뮬레이션 하였다. 시뮬레이션에 사용된 카메라 성능은 아래 표 3과 같다. 카메라 B는 카메라 A보다 시야거리가 1.5배 길지만, 시야각은 0.75배 작다. 카메라 C는 시야거리가 카메라 A보다 2배 길지만 비용은 2.5배 크다.

(표 3) 카메라 성능

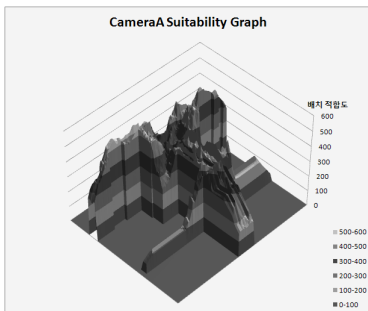
카메라	시야거리(그리드)	시야각	비용
A	60	60°	80
B	90	45°	120
C	120	60°	200

그리드 알고리즘기반의 배치 알고리즘을 이용하여, 세 가지 카메라에 대한 적합도를 계산하였다. 카메라 적합도 그래프는 그림 10과 같다.

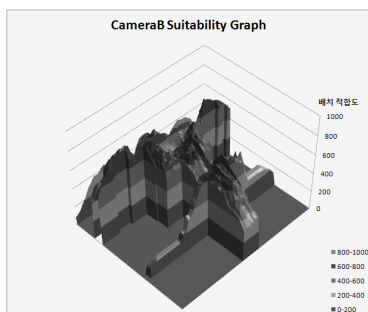


(다) 카메라C의 적합도 그래프
(그림 10) 카메라 적합도

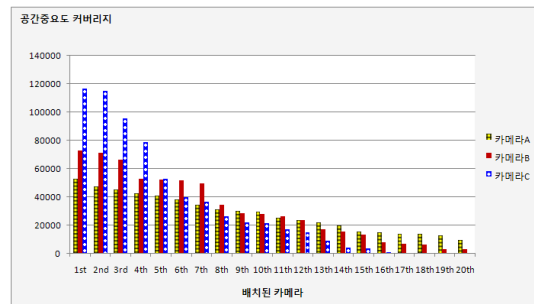
적합도 그래프의 x와 y축은 모델링된 2차원 공간의 좌표이다. 적합도의 값이 높은 순서대로 카메라를 배치하여 최적의 카메라 배치에 대한 조건이 만족할 때까지 배치를 반복한다. 그림 11은 이러한 방법으로 20개의 카메라를 배치를 할 때 공간 중요도 커버리지의 변화를 보이고 있다.



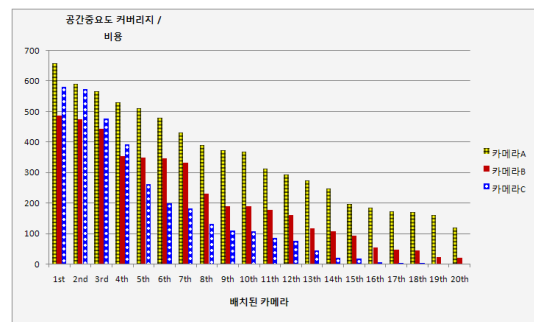
(가) 카메라A의 적합도 그래프



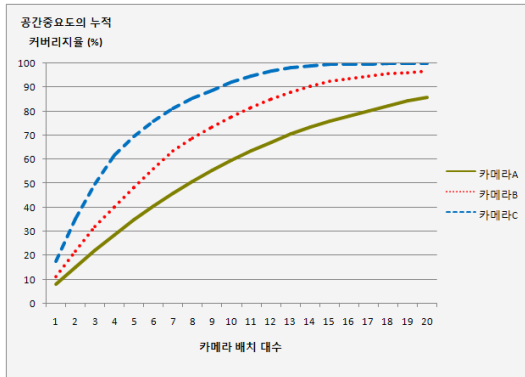
(나) 카메라B의 적합도 그래프



(가) 카메라 배치 순서에 따른 공간중요도 커버리지



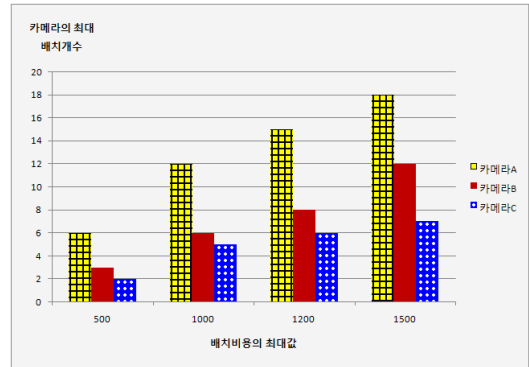
(나) 카메라 배치 순서에 따른 비용대비 공간중요도 커버리지



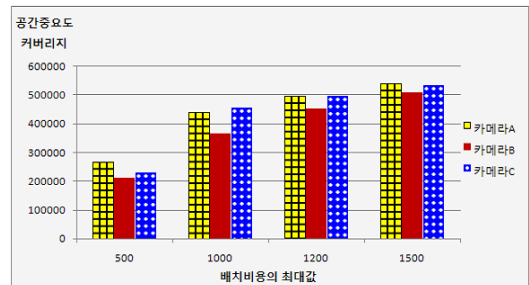
(다) 카메라 배치개수에 따른 공간중요도의 누적 커버리지를 (그림 11) 카메라 배치에 따른 커버리지 변화

그림 11-(가) 와 (나)에서 보는 바와 같이, 카메라 C의 경우, 단위개수당 공간 중요도 커버리지가 초기 배치에는 가장 높지만, 비용당 공간 중요도 커버리지는 카메라 배치 개수가 증가할수록 낮아졌다. 이것은 건물의 내부에는 벽과 같은 영역이 존재하기 때문에 카메라의 시야거리가 길고, 시야각이 넓어도 불필요하기 때문이다. 또한, 그림 11-(다)에서 보는 바와 같이, 적합도가 높은 순서대로 카메라가 배치되므로, 공간 중요도 커버리지 비율은 처음에는 높은 증가율을 보였지만, 점차 증가율이 낮아지는 것을 확인할 수 있었다. 예를 들어, 카메라 C는 15 대 이후의 공간 중요도 커버리지는 단위 개수당 0.08% 이하였다.

시뮬레이션의 종료 조건으로 카메라 배치 한계 비용을 500, 1000, 12000, 1500으로 정하여 세 가지 카메라에 대해 각각 시뮬레이션 하였다. 한계 비용 내에서 카메라 A, 카메라 B, 카메라 C의 최대의 배치개수를 구하고, 그 배치개수에서 최대 공간중요도 커버리지의 최대값을 계산하여 그 중 가장 높은 공간중요도 커버리지를 나타낸 카메라를 선택하였다. 시뮬레이션을 통해 구한 최적의 카메라 개수와 커버리지에 대한 결과는 그림 12 와 같다.

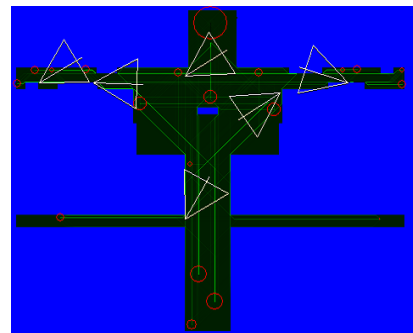


(가) 한계비용에 대한 카메라의 최대 배치개수

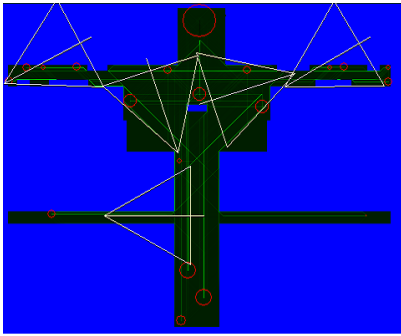


(나) 한계비용에 대한 카메라 타입별 최대 공간중요도 커버리지 (그림 12) 한계비용에 대한 카메라 최대 배치 개수와 최대 공간중요도 커버리지

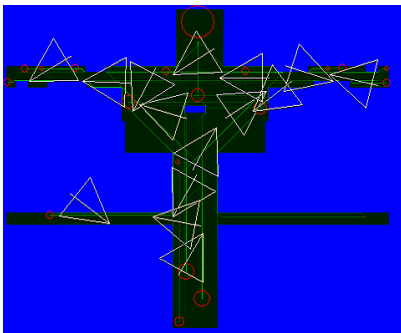
그림 12에서 보는 바와 같이, 한계 비용이 500, 1200, 1500일 때 카메라 A가 공간 중요도 커버리지가 가장 높았으며, 한계비용이 1000일 때 카메라 C가 가장 높았다. 이러한 최적의 카메라 개수와 커버리지에 대해 카메라 배치에 대한 시뮬레이션 결과는 그림 13과 같다.



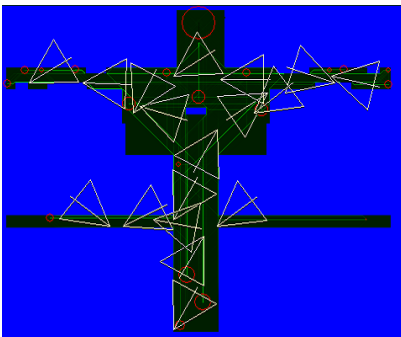
(가) 한계비용이 500 일때



(나) 한계비용이 1000 일때



(다) 한계비용이 1200 일때



(라) 한계비용이 1500 일때

(그림 13) 한계비용에 대한 최적의 카메라 배치위치

그림 13에서 삼각형은 카메라의 FOV이며, 삼각형내의 선은 카메라의 방향이다. 삼각형의 두 변과 삼각형 내에 직선이 만나는 점이 카메라의 위치이다. 카메라의 배치는 사람이 이동할 가능성이 높고 여러 경로들이 중첩되지 않는 홀 지역보다는 좁은 통로가 공간중요도가 높게 집중되었다. 또한, 카메라 형태에 따라 우선으로 배치되는 위

치가 달랐다. 넓은 홀을 2~3대 만으로 커버할 수 있는 카메라 C의 경우 카메라 A와 카메라 B와 달리 좁은 통로보다는 경로들이 분산되어 있지만 거의 모든 경로들이 통과하는 위치인 홀에 우선으로 카메라들이 배치되는 것을 알 수 있었다.

6. 결 론

본 논문은 효과적인 영상 감시 시스템 구축을 위하여 에이전트를 기반으로 최적의 카메라 배치에 대해 시뮬레이션 하였다. A* 알고리즘기반으로 사람의 이동을 추론하여 에이전트를 모델링하고 구현하였다. 또한, A* 알고리즘으로 추출한 경로 주변의 이동 가능한 공간 중 사람이 이동할 수 있는 영역을 추론하였으며, 공간중요도 레이어에 중요도를 재계산하여 경로확장 알고리즘을 구현하였다. 실제 공간에 대한 공간의 구조와 사람의 유동량을 측정하여 확률모델로 이용하였으며, 카메라의 성능과 비용을 기반으로 시뮬레이션하여 최적의 카메라 배치위치를 결정하였다.

향후의 연구로는 도시단위의 공간에서 다양한 성능의 카메라들을 조합하여 설치하는 경우에 효율적인 카메라 배치가 가능하도록 최적화하는 하는 것이다.

참 고 문 헌

- [1] Computing," *IEEE Computer*, October 1993, vol. 26, no. 10, pp. 71-72.
- [2] J.J. Barton, and V. Vijayaraghavan, "UBIWISE, A Simulator for Ubiquitous Computing Systems Design, " Technical Report HPL-2003-93, *HP Laboratories*, Palo Alto, 2003.
- [3] E. O'Neill, M. Klepal, D. Lewis, T. O'Donnell, D. O'Sullivan, and D. Pesch, "A Testbed for Evaluating Human Interaction with Ubiquitous Computing Environments," *Proc. of 1st Int'l Conf. on Testbeds and Research Infrastructures*

- for the Development of NeTworks and Communities*, 2005, pp. 60-69
- [4] H. Nishikawa, S. Yamamoto, M. Tamai, K. Nishigaki, T. Kitani, N. Shibata, K. Yasumoto, and M. Ito, "UbiREAL: Realistic Smartspace Simulator for Systematic Testing," *Proc. of the 8th Int'l Conf. on Ubiquitous Computing (UbiComp2006)*, LNCS 4206, Sep. 2006, pp. 459-476.
- [5] Miquel Martin, Petteri Nurmit, "A Generic Large Scale Simulator for Ubiquitous Computing," *Proc. of 3rd Int'l Conf. on Mobile and Ubiquitous Systems: Networking & Services*, 2006, pp. 1-3.
- [6] S. Ram, K. R. Ramakrishnan, P. K. Atrey, V. K. Singh and M. S. Kankanhalli, "A design methodology for selection and placement of sensors in multimedia surveillance systems," *Proc. of the 4th ACM Int'l workshop on Video surveillance and sensor networks*, 2006, pp. 121-130.
- [7] E. Horster and R. Lienhart, "On the optimal placement of multiple visual sensors," *Proc. of the 4th ACM Int'l workshop on Video surveillance and sensor networks*, 2006, pp. 111-120.
- [8] J. Zhao and S.-c.S. Cheung, "Multi-camera surveillance with visual tagging and generic camera placement," *Proc. of 1st ACM/IEEE Int'l Conf. on Distributed Smart Cameras*, 2007, pp. 259-266.
- [9] Nils J. Nilsson. "Artificial Intelligence : A New Synthesis," *Morgan Kaufmann Publishers*, 1998.
- [10] J. Pearl, "Heuristics: Intelligent search strategies for computer problem solving," *Addison-Wesley*, 1984.
- [11] P. Ballard, F. Vacherand, "The Manhattan Method: A Fast Cartesian Elevation Map Reconstruction from Range Data," *Proc. of IEEE Conf. on Robotics and Automation*, 1993, pp. 143-148.

● 저 자 소 개 ●



변 유 인 (Uin Burn)

2007년 아주대학교 전자공학과 졸업(학사)
2009년 아주대학교 대학원 전자공학과 졸업(석사)
2009년~현재 (주)세나테크놀로지 주임연구원
관심분야 : 유비쿼터스 컴퓨팅, 패턴인식, 컴퓨터 비전, 시뮬레이션
E-mail : byunja@gmail.com



남 윤 영 (Yunyoung Nam)

2001년 아주대학교 정보 및 컴퓨터공학과 학사
2003년 아주대학교 정보통신전문대학원 정보통신공학과 석사
2007년 아주대학교 정보통신전문대학원 정보통신공학과 박사
2007년 아주대학교 유비쿼터스시스템연구센터 선임연구원
2009년~현재 뉴욕주립대(Stony Brook) 전자 및 컴퓨터공학과 (Post Doc.)
2010년~현재 아주대학교 유비쿼터스컨버전스연구소 연구교수
관심분야 : 데이터베이스, 멀티미디어 시스템, 정보 통합, 시뮬레이션, 영상처리, 패턴인식, 컴퓨터 비전, 유비쿼터스 컴퓨팅
E-mail : young022@gmail.com



조 위 덕 (We-Duke Cho)

1981년 서강대학교 전자공학과 학사
1983년 한국과학기술원 전기 및 전자공학과 석사
1987년 한국과학기술원 전기및전자공학과 박사
1990년 금성전기(현 LG전자) 기술연구소 DSP 연구실장
1991년 전자부품연구원 시스템연구본부 본부장
2003년~현재 정보통신부 21세기프론티어사업 (재)유비쿼터스컴퓨팅사업단장
2004년~현재 아주대학교 전자공학부 교수
아주대학교 유비쿼터스시스템연구센터장
관심분야 : u-지능공간시스템개발, 스마트임베디드오브젝트개발(스마트베드, 스마트테이블, 스마트 미러, 스마트 카메라 등
e-mail: wdukecho@gmail.com