

## 자율 주행 로봇을 위한 네트워크 기반의 능동적 시스템 구조의 설계

이현구\*, 이병주\*, 서삼준\*\*, 박귀태\*\*\*  
 고려대학교 전기전자전파공학부 자동제어연구실\*  
 안양대학교 전기전자공학과\*\*  
 고려대학교 전기전자전파공학부\*\*\*

### Design of active system architecture for autonomous mobile robot under network environments

Hyun-Gu Lee\*, Byoung-Ju Lee\*, Sam-Jun Seo\*\*, Gui-Tae Park\*\*\*  
 Automatic Control LAB, Dept. of Electrical, Electronic, Radio Engineering, Korea University\*  
 Dept. of Electrical & Electronic Engineering, Anyang University\*\*  
 Dept. of Electrical, Electronic, Radio Engineering, Korea University\*\*\*

**Abstract** - 현재까지의 자율 주행 로봇에 대한 연구는 완전히 지능화된 로봇으로 작업을 수행하는 연구가 대부분이었다. 자율 주행 로봇이 완전한 지능을 지니기 위해서는 고비용, 고성능의 센서가 필요하다. 본 논문은 지능화 공간의 개념을 사용하여 네트워크 기반의 자율 주행 로봇을 구동하기 위한 능동적인 시스템 구조를 제안하며 고성능의 센서가 없어도 효과적인 주행을 함을 보인다. 본 논문에서 제안하는 시스템 구조에서 네트워크 상의 여러 가지 자원들(로봇, 비전 시스템, 디스플레이 시스템)은 서비스가 되어 서로의 기능을 능동적으로 사용할 수 있다. 네트워크 상에서 각 서비스를 공유하기 위하여 JAVA기반의 프로토콜 Jini를 사용하였다. 본 논문에서 제안한 자율 주행 로봇은 주변 환경 및 주어진 작업에 따라 기능적으로 모듈화된 서비스를 환경상의 여러 자원들과 공유하여 운용된다.

사건에 대한 정보를 공유할 수 있다. Coen은 Scatterbrain[1]이라는 개념의 지능화 공간 상의 에이전트에 대한 연구를 수행하였다. 지능화 공간 상에서 에이전트의 임무는 공간상의 모든 자원(물체 추적 시스템, 음성 인식 시스템 등등)들을 하나로 묶고 내부의 정보를 하나로 통합하는 것이다. 여기에서는 지능화 공간을 Smart Room이라는 용어를 사용한다. 동경대의 Hashimoto lab에서는 이러한 지능화 공간상에 자율 주행 로봇을 추가하려는 시도를 하고 있다.[2]

## 1. 서 론

본 논문에서는 네트워크 환경 하에서 자율 주행 로봇을 적용할 때 자율 주행 로봇과 네트워크에 연결된 자원 사이에 서로의 기능을 공유하여 이를 능동적으로 사용하는 방법을 제시한다. 지금까지의 자율 주행 로봇에 대한 연구는 자율 주행 로봇 자체의 지능화를 이용하기 위한 것이 대부분이었다. 이를 위해 자율 주행 로봇에 고비용, 고성능의 센서(레이저 레인지 파인더, 자이로스코프 등)들을 사용하기 때문에 일반적으로 사용할 수 있는 자율 주행 로봇을 만들기가 어려운 것이 지금의 현실이다. 최근에 이러한 문제점을 해결하기 위한 연구가 많이 진행되고 있는데, 이 중에서도 지능화 공간의 개념이 가장 설득력 있는 방안으로 제시되고 있다.

본 논문에서는 지능화 공간의 개념을 사용해 완전 지능화된 자율 주행 로봇이 아니더라도 주어진 임무를 수행할 수 있음을 보인다. 지능화 공간 상에서 자원들간의 정보를 교환하기 위한 방법으로 네트워크가 필요하게 되며 자율 주행 로봇은 지능화 공간 상에서 네트워크를 통해 여러 종류의 자원을 능동적으로 사용하게 된다.

## 2. 본 론

### 2.1 지능화 공간

지능화 공간은 공간 자체를 모델링할 수 있고 또한 공간 상에서 일어나는 사건들을 인지할 수 있는 시스템이다. 지능화 공간 상에서 일어나는 사건을 관찰하고 학습함으로써 스스로를 모델링한다. 예를 들어 공간 상에서 인간이 활동하는 동선을 관찰함으로써 자율 주행 로봇이 움직일 수 있는 동선의 일부를 찾을 수 있고 공간 상에서 일어나는 사람의 말이나 행동을 사건으로 인지하여 이를 자율 주행 로봇과 같은 기계와 사람의 의사소통의 수단으로 사용할 수 있다. 또한 지능화 공간은 네트워크를 사용함으로써 사람, 기계 그리고 공간상의 모든 개체가

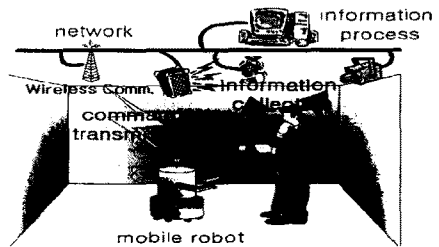


그림 1. 지능화 공간

### 2.2 네트워크를 이용한 자율 주행 로봇 시스템

#### 2.2.1 자율 주행 로봇 시스템의 구조

일반적으로 자율이동로봇이 주어진 환경 하에서 원하는 작업을 수행하기 위해서는 로봇을 구성하는 소프트웨어 요소들이 구조적으로 어떻게 이루어지고 요소들간 어떻게 상호 작용할 것인지를 명확히 나타내는 제어기의 적절한 구조가 필요하다. 이러한 제어기를 설계하는 방법 중에 가장 많이 연구된 것이 계층적 구조(hierarchical architecture), 행동 기반 구조(behavior based architecture), 그리고 복합형 구조(hybrid Architecture)가 있다.

계층적 구조는 작업 환경이 어떠한 이유에 의해 시간에 따라 변화하는 환경 하에서는 적용하기가 어렵다.

계층적 제어 구조의 단점을 극복하여 작업 환경의 변화에 잘 적응하는 모델이 행동기반 구조이다. 센서 계층과 액추에이터 계층이 바로 연결되어 있어 센서의 입력에 곧바로 반응할 수 있는 구조를 가진다. 이렇게 함으로써 환경에서 보다 빠른 자율 주행 로봇의 응답을 얻을 수 있다. 그러나 어떠한 행동을 선택해야 하는지에 관한 적절한 중재를 하기가 어려워 규모가 큰 제어 시스템을 만들기에는 부족한 면이 있다.[3][4][5]

복합형 구조는 서로 다른 목적을 수행하기 위하여 선형적인 환경에 대한 정보를 사용하고 변화하는 작업 환경에 대하여도 적절히 반응하게 하는 구조이다.

### 2.2.2 네트워크를 사용한 자율 주행 로봇 시스템의 구조

일반적인 로봇 시스템에서 네트워크를 사용하는 이유는 크게 세 가지를 들 수 있다. 첫 번째는 다 개체 로봇 시스템에서는 자율 주행 로봇간의 상호 작용을 통하여 수행하고자 하는 목적을 좀더 지능적이고 효과적으로 달성하기 위해서이다. 두 번째는 자율 주행 로봇을 원격지 조정하는 수단으로써 네트워크가 사용된다. 마지막 세 번째는 지능화 공간에서 공간상의 모든 자원들을 공유하기 위한 수단으로 네트워크가 사용된다.

다 개체 로봇시스템이나 지능화 공간에서 네트워크를 사용하기 위해서 시스템을 구현하는 방법에는 크게 두 가지가 있다. 즉, 네트워크의 관점에서 자율 주행 로봇과 네트워크에 연결된 다른 자율주행 로봇 또는 자원들을 동등한 입장에서 보느냐 아니냐에 따라 중앙 집중적 구조와 분산형 구조로 나눌 수가 있다.

중앙 집중적 구조는 서버/클라이언트 구조와 유사하며 서버의 기능을 하는 개체에서 네트워크를 사용하는 모든 자원들을 관리, 분배하는 방법이다. 즉, 서버는 이러한 시스템에서 가장 중요하고 핵심적인 역할을 담당한다. 그러나 이러한 중앙 집중적인 구조는 네트워크 장애나 결함 같은 이유로 문제가 생겼을 경우를 대비하기가 어려우며 서버의 기능을 확장하거나 새로운 시스템을 추가하는 경우에는 서버 기능을 하는 로봇에 많은 수정을 가해야 하는 문제점이 있다.

이에 반해 분산형 구조는 네트워크 상에서 각각의 자원들을 분산시키는 개념으로 서버/클라이언트로 나누는 것이 아니라 각각의 자원이 서버가 되고 클라이언트가 되는 것이다. 이렇게 함으로써 각각의 자원들은 동등한 입장에서 다른 자원을 사용할 수 있게 된다. 분산형 구조에서 개체나 자율 주행 로봇을 확장하면 작업을 수행하는데 변동사항이 있는 자율 주행 로봇만을 다시 설계하면 된다. 또한 소수의 자율 주행 로봇에 결함이 생길 경우에 대비하기 쉽다.

### 2.3 네트워크 기반의 능동적 시스템 구조

네트워크 기반의 자율 주행 로봇을 위한 시스템 구조를 설계하기 위해서는 자율 주행 로봇을 이루는 중요한 요소와 네트워크에 연결된 자원들을 간에 어떠한 방법으로 정보를 공유하고 사용할 것인지를 결정하는 것이 중요하다.

자율주행로봇을 설계할 때 앞에서 설명한 계층적 구조, 행동기반 구조, 그리고 복합형 구조를 생각할 수 있다. 세 가지 구조에서 공통적으로 사용할 수 있는 항목은 환경을 인식하는 부분, 로봇을 구동하는 부분, 그리고 자율주행로봇을 어떻게 운용하는가에 대한 알고리즘 부분으로 나눌 수 있다. 본 논문에서는 자율 주행 로봇의 이 세 가지 부분을 객체 지향적인 프로그래밍 기법으로 서로 다른 객체로 취급하여 네트워크 환경에서 사용하기 위한 구조를 제시한다. 자율 주행 로봇 이외의 자원들은 각각의 객체로 취급하여 네트워크 상에서 자율주행로봇이 동작 하게 되었을 때, 모든 자원들을 자율주행로봇이 주체가 되어 사용할 수가 있다.

객체 개념의 구조를 갖는 시스템을 구현하려면 객체지향적 언어를 사용하는 것이 설계단계의 개념을 그대로 구현 단계로 가져갈 수 있다. 객체지향적 구조를 설계하기 위한 언어에는 C++, JAVA 같은 언어 등이 주로 사용된다. 선(SUN)에서 제안한 JINI는 JAVA 언어를 기반으로 객체 지향적 구조를 가지며 분산 환경에서 적합하게 사용할 수 있게 만든 API 및 통신 프로토콜을 말하는데 네트워크 환경에서 서비스 개념의 자원을 수행하고 네트워크에 관련된 유지, 보수, 관리의 문제를 쉽게 해결할 수 있는 구조를 가지고 있다.

### 2.4 JINI를 이용한 자율 주행 로봇과 자원의 객체화 방법

#### 2.4.1 JINI의 구조

JINI는 네트워크 상에서 장치나 서비스들을 찾고, 추가하고 또는 제거하는 등의 작업을 손쉽게 할 수 있는 방법을 제공한다. 또한 프로그래머들이 장치들 간의 상호 연결을 편하게 수행 할 수 있도록 프로그래밍 구조 및 모델을 제공한다.

그림 3은 JINI서비스를 사용하기 위한 과정을 보여주고 있다. 먼저 서비스 제공자는 디스커버리(discovery) 과정을 통해 룩업서비스(lookup)를 찾고 찾아진 룩업서비스에 자신의 서비스를 조인(join) 과정을 통해 등록하게 된다. 서비스를 등록하면 자신의 서비스를 나타내는 서비스 대행자(service proxy)가 룩업 서비스에 존재하게 된다. 만약 네트워크 상에 서비스(service)를 사용하려는 클라이언트가 등장하게 되면 클라이언트는 역시 룩업서비스를 찾고 서비스 대행자를 다운로드(download) 받은 뒤 서비스 제공자와 1:1통신을 통한 서비스를 사용한다.

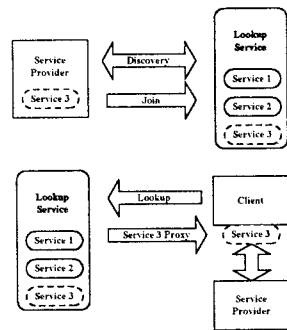


그림 3 JINI서비스의 사용

#### 2.4.2 JINI를 이용한 자율 주행 로봇과 자원의 객체화 방법

JINI를 사용하여 홈 네트워크 환경의 자원과 자율 주행 로봇을 객체화하기 위한 방법은 위에서 객체로 나누어진 자율 주행 로봇과 네트워크 환경상의 자원을 JINI 서비스로 만드는 것이다. 이렇게 JINI 서비스로 만들어진 모든 객체는 서로 다른 객체를 서비스의 개념으로 사용할 수 있게 되고 동시에 그 서비스를 사용하는 클라이언트가 된다. 여기서 자율 주행 로봇은 주로 다른 자원의 서비스를 사용하는 클라이언트의 입장이 된다. 네트워크 환경 하에서 네트워크에 연결된 모든 자원을 JINI 서비스화 하고 새로운 자원이 네트워크에 추가 될 경우 새로 추가된 자원도 이미 네트워크 상에서 존재하는 자원을 사용할 수 있고 또한 자율 주행 로봇도 새로 추가된 자원을 사용할 수 있게 된다.

그림 4는 JINI를 이용하여 네트워크를 이용하는 자율 주행 로봇의 시스템 구조를 만든 것이다. 네트워크 환경에서 자율 주행 로봇은 세 가지의 서비스로 구성되고 네트워크상에 연결된 기타 자원들은 각각의 기능을 수행하는 하나의 서비스로 구성된다. 여기서 자율 주행 로봇은 크게 자율 주행 로봇 운용 알고리즘 서비스, 센싱 서비스 그리고 액추에이터 서비스로 나눌 수 있다.

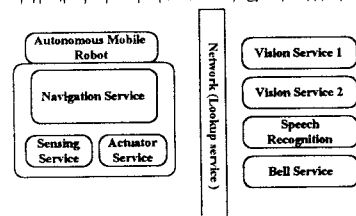


그림 4. JINI 서비스화한 예

센싱 서비스는 자율 주행 로봇상에 존재하는 것이 아니라 지능화 공간상에서 환경을 인식하는 여러 가지 서비스(비전 서비스 등)으로 대체될 수 있다. 또한 액추에이터 서비스는 로봇의 행동을 결정하는 중요한 서비스이기 때문에 좀 더 빠른 작업 속도를 원한다면 자율 주행 로봇 운용 알고리즘 서비스 안에서 집적 관리할 수도 있다.

JINI 서비스는 성격에 따라 크게 두 가지의 종류로 나뉜다. 하드웨어와 같은 물리적인 것을 서비스화 한 것과 소프트웨어 같은 논리적인 것을 서비스화 한 것으로 나뉠 수 있다. 그림 5에서는 자율 주행 로봇을 위해 필요한 서비스 중 몇 가지의 물리적 서비스와 논리적인 서비스를 보여 주고 있다. 자율 주행 로봇은 정해진 작업을 수행할 수 있는데, 이러한 작업은 JINI 서비스를 조합하고 단순히 사용함으로써 수행될 수 있다. 작업을 전환하는 것도 또한 서비스 또는 서비스의 조합에 의해 이루어진다.

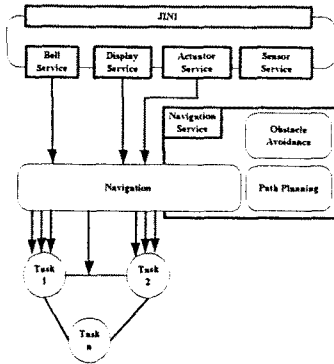


그림 5. JINI서비스간의 관계

그림 5의 구조에서 논리적인 서비스는 자율 주행 로봇의 주행을 담당하는 주행 서비스( navigation service )이다. 주행 서비스는 여러 가지 물리적인 서비스를 사용해 자율 주행 로봇이 하고자 하는 여러 가지 작업을 수행하기 위한 알고리즘을 수행하고 있다.

그림 6은 전체적인 시스템에서의 각 서비스간의 관계를 나타내고 있다. 자율 주행 로봇은 벨 서비스를 사용해 이벤트를 관리하며 이러한 이벤트는 자율 주행 로봇이 다음 목표점을 설정하는 데 중요한 역할을 수행한다. 이벤트가 발생하면 이벤트 관리자에 이벤트가 저장된다.

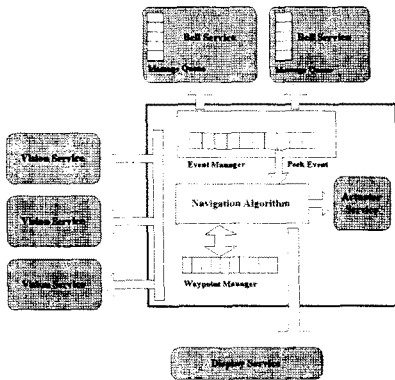


그림 6. 전체 시스템 구조

자율 주행 로봇의 주행 알고리즘에서는 이벤트 관리자에 있는 이벤트를 확인하고 현재 상태를 결정하게 되며 이 상태에 따라 목표점을 설정하게 된다. 자율 주행 로

봇의 목표점은 목표점을 순서대로 입력해 놓은 목표점 관리자( waypoint manager )에 저장되어 있으며 주행 알고리즘이 이 목표점들을 순서대로 주행하게 된다. 또한 어떠한 목표점에 도달하거나 이벤트가 일어났음을 감지할 경우 디스플레이 서비스를 사용하여 이러한 정보들을 표시한다.

## 2.5 실험

### 2.5.1. 실험 환경

그림 7은 전체적인 시스템 구성을 보여주고 있다. 각각의 서비스는 유무선의 네트워크를 통해 연결되며 JAVA기반의 JINI서비스로 구현하기 위하여 IBM PC 와 그와 관련된 하드웨어로 구성되어 있다.

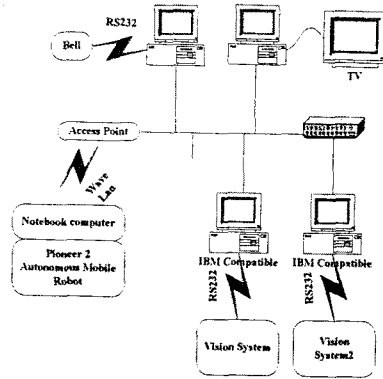


그림 7. 전체 실험 환경

자율 주행 로봇에 관한 JINI서비스는 움직여야하는 특성 때문에 유선의 네트워크는 사용할 수 없다. 자율 주행 로봇에 탑재되어 있는 노트북 컴퓨터가 실제로 로봇의 액추에이터를 구동하는 프로그램을 탑재하고 노트북 컴퓨터를 무선 랜 시스템에 연결하였다. 나머지 서비스들은 유선 네트워크를 사용하여 각각의 기능을 수행하게 하였다. 본 실험에 사용한 자율 주행 로봇은 ActiveMedia에서 제작한 Pioneer2 이다.

비전 시스템은 기본적으로 CTS( color tracking system )을 사용한다. 자율 주행 로봇 위에 컬러 마커( color marker )를 부착한 뒤 CTS를 사용하여 이를 추적하면서 자율 주행 로봇의 위치를 확인하게 된다. 본 실험에서는 기본적으로 자율 주행 로봇에 한 가지 색의 컬러 마커만을 사용한다.

그림 8에서 보는 바와 같이 카메라는 공간의 벽면에 부착되어 바닥을 관찰하게 된다.

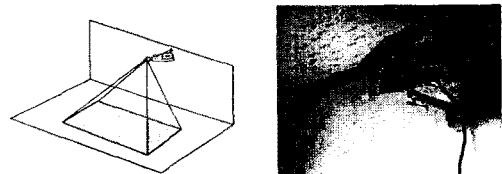


그림 8. 비전 시스템

벨 시스템은 이벤트를 발생시킬 때 발생하며 본 실험에 사용된 벨 시스템은 크게 두 가지로 나뉜다. 첫 번째는 자율 주행 로봇을 집적 호출하여 지정된 위치로 이동시키는 기능을 수행하고 두 번째는 자율 주행 로봇이 수행하는 작업을 전환시키는 기능이 있다.

디스플레이 시스템은 전체 시스템에서 자율 주행 로봇의 위치를 TV에 표시해 준다. 또한 벨 이벤트가 발생한 위치를 보여주고 자율 주행 로봇의 다음 목표 지점을 표

시해 준다.

본 실험에서 사용한 비전 시스템과 벨 시스템 그리고 자율 주행 로봇은 하드웨어를 직접 제어해야 하기 때문에 JAVA로 곧바로 구현하지 못하는 제약이 있다. 이러한 문제를 해결하기 위해서 이러한 하드웨어를 제어하는 Windows용 프로그램을 작성하고 UDP 통신을 사용해 JAVA 프로그램과 정보를 교환하는 방식을 사용한다.

### 2.5.2 실험

실험 환경은 그림 9에서와 같이 두 개의 인접한 직각 사각형의 영역 두 개로 이루어져 있다. 각각의 영역은 서로 다른 비전 시스템에 의해서 관찰되며 자율 주행 로봇 위에 부착된 컬러 마커의 색을 계속 추적하면서 자율 주행 로봇의 위치를 결정한다.

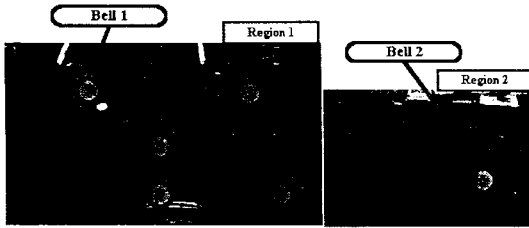


그림 9. 실험 영역

첫 번째 실험은 자율 주행 로봇이 그림 8에 나타난 목표점들을 1-2-3-4-5-6-1의 순서로 설정하였을 때 비전 서비스 2개만을 사용해 주행하는 것을 실험하였다.

표 1은 자율 주행 로봇이 1-2-3-4-5-6의 구간을 순서대로 주행하였을 때 각 지점에서 원래 자율 주행 로봇이 있어야 할 위치와 실제로 자율 주행 로봇이 위치한 곳과의 거리를 절대값으로 나타낸 것이다.

표 5.1 위치 오차(cm)

position	1	2	3	4	5	6
1	25	15	10	5	15	20
2	10	25	15	10	10	25
3	15	5	15	10	20	20
4	25	5	20	15	15	10
5	15	10	20	20	5	15
6	5	25	15	10	20	5
7	10	15	5	20	15	25
8	30	20	20	15	5	30
9	30	5	25	10	20	25
10	25	30	15	5	0	30
average	19	15.5	16	12	12.5	20.5

1-2구간의 오차가 큰 이유는 1-2구간이 다른 구간에 비해 카메라로부터 먼 거리에 위치하고 있기 때문이다. 또한 5-6 구간은 주행 구간이 다른 구간에 비하여 비교적 길어 오차가 크다.

두 번째 실험은 모든 서비스를 사용하여 자율 주행 로봇이 주행하는 것이다. 우선 자율 주행 로봇이 할 수 있는 일을 두 가지로 정한다. 이 두 가지 일은 벨 서비스에 의하여 활성화된다.

1. 자율 주행 : 로봇이 사람이 간섭하거나 이벤트가 일어났을 때를 제외한 경우에 지능화 공간상에서 지정된 위치를 자율 주행하는 경우

2. 이벤트 반응 : 벨 서비스가 실행되었을 경우 벨이 눌린(이벤트가 발생한 위치) 위치로 주행하는 경우

그림 10은 이벤트를 임의의 순서대로 발생시키고 그때의 자율 주행 로봇의 주행 경로를 2회 관찰한 것이다.

먼저 벨 1에서 이벤트가 발생하고 그 다음 자율 주행을 들어가는 이벤트가 발생하게 하였다. 자율 주행 로봇이 영역 1의 네 모서리를 주행하였을 경우 다시 벨 2에서 이벤트가 발생하게 되고 벨 2의 위치로 이동하게 된다. 그림 10에서 보듯이 자율 주행 로봇이 큰 오차 없이 주행하고 있음을 알 수 있다.

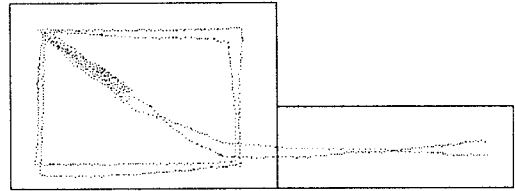


그림 10. 주행 결과

### 3. 결 론

본 논문에서는 네트워크 환경 하에서 사용하기 위한 자율 주행 로봇의 능동적인 시스템 구조를 제안하였다. 자율 주행 로봇의 완전 지능화를 위해서는 고성능의 센서가 필요하게 되는데 본 논문에서 직진과 방향 전환만을 하는 구조가 간단한 자율 주행 로봇에 비전 서비스를 사용하여 주행하는 실험을 하였다. 또한 자율 주행 로봇이 자신이 사용할 수 있는 자원을 네트워크 상에서 능동적으로 찾아 사용하기 때문에 향후 다른 서비스를 추가할 경우 로봇 주행 알고리즘만을 변경하면 곧바로 새로운 시스템을 만들 수 있는 가능성을 확인하였다.

자율 주행 로봇의 위치를 추적하기 위해 네트워크를 통해 비전 서비스를 수행하기 때문에 약간의 시간 지연 현상이 발생하였다. 이 문제는 JINI의 수행 속도가 상대적으로 느리기 때문이며 이를 해결하기 위해 빠른 수행 속도가 필요한 부분은 JINI 외에 방법을 사용할 계획이다. 또한 향후 좀 더 다양한 서비스를 본 시스템에 추가함으로써 최종적으로 홈 네트워크 환경 하에서 사용할 수 있는 플랫폼을 구축할 계획이다. 이와 더불어 체계적인 서비스 구조를 확립하여 새로운 서비스를 추가할 때 미리 정해진 인터페이스를 사용하게 하여 표준화된 환경을 마련하는 것도 중요하다. 지금 실험 환경은 JINI 서비스를 실행하기 위해 PC를 사용하였다. 이러한 문제를 해결하기 위해 Embedded Java System을 사용하는 것이 필요하다.

### [참 고 문 헌]

- [1] Coen, M. "SodaBot: A Software agent Environment and Construction System. AI Lab Technical Report 1493", Massachusetts Institute of Technology, Cambridge, MA, 1994.
- [2] J.H. Lee, G. Appenzeller, H. Hashimoto, "Physical agent for a sensed, networked and thinking space", In Proc. of IEEE Int. Conf. On Robot and Automation, 1998.
- [3] R. C. Arkin, "Behavior-based robotics", MIT Press, 1998
- [4] R. A. Brooks, "Intelligence without Representation", Artificial Intelligence, Vol. 47, pp. 139-159, Jan., 1991
- [5] J. M. Park, "A Hybrid Control Architecture Using a Reactive Sequencing Strategy for Mobile Robot Navigation", International con. on Intelligent Robots and Systems
- [6] B.J. Lee, G.T. Park, "A Design of a Physical Agent for a Cooperative Space", Proc. Of KACC99, vol. B, pp. 168-171, 1999
- [7] B.J. Lee and G. T. Park, "A Robot in Intelligent Environment : Soccer Robot", Proceedings of the 1999 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pp. 73-78, 1999
- [8] W. Keith Edwards, "Core JINI", PTH PH, 1999