

論文 2001-38CI-3-2

가상세계 구축을 위한 체계적 환경 모델링 방법

(A Systematic Model for Constructing Environment in Virtual World)

金魯淳*, 朴宗熹*

(No-Sun Kim and Jong-Hee Park)

요약

본 논문에서는 가상세계를 체계적으로 구축하는데 기반이 되는 가상환경을 설계하고 구현한다. 지금까지 가상환경은 가상현실기술을 사용한 여러 시스템에서 구현되었지만 가상환경의 설계는 체계화된 구조화 방법 없이 이루어져 왔다. 본 논문에서는 실세계를 구성하는 환경변수의 분류를 바탕으로 가상환경을 기능과 범위에 따라 6개의 계층으로 구조화한다. 이 구분된 각 계층들은 각기 독립적으로 동작하며, 주기적으로 정보를 교환하여 전체환경을 구성하게 된다. 설계된 가상환경은 실세계 환경과 같이 다양한 상황을 보여주며 확장에 있어서도 유연함을 보여 준다. 또한 구조화된 환경에서 동작하는 중요한 환경변수들의 효율적 구현방법을 제시한다. 제안된 가상환경의 구조화 방안을 바탕으로 가상세계를 구현하여 시험함으로써 효용성을 보인다.

Abstract

In this paper, we design and implement a virtual environment which forms the basis of constructing a virtual world in a systematic fashion. Though virtual environments have been developed in several systems using the virtual reality, they are designed without a systematic method of constructing. In our framework, a virtual environment is structured in six layers according to their functions and ranges based on a classification of environmental factors of which the real world is composed. These layers compose a universal environment in our virtual world by operating independently and exchanging information with one another periodically. The designed environment shows that it can provide diverse real-life situations and is easily extensible to enhance its functionalities. We present effective design schemes for several important environmental factors. We demonstrate viability of our framework through implementing a plausible virtual world based on the proposed schemes.

I. 서론

가상현실이란 현실에 구애받지 않는 상상의 세계를

* 正會員, 慶北大學校 電子工學科
(School of Electronic & Computer Engineering,
Kyungpook National University)

※ 본 연구과제는 정보통신부의 대학정보통신연구센터
육성 지원사업의 예산으로 수행된 것임을 밝힙니다.
接受日字: 2000年10月2日, 수정완료일: 2001年4月20日

컴퓨터를 사용하여 현실과 같이 만들어 내어 인체의 모든 감각기관(눈, 귀, 피부, 코, 입)이 인위적으로 창조된 세계에 몰입(immersion)됨으로써 자신이 바로 그곳에 있는 것처럼 느낄 수 있는 가상공간의 세계이다. 현재 가상현실 기술은 컴퓨터 성능의 향상에 힘입어 컴퓨터 그래픽, CAD, GIS, 멀티미디어, 의학, 게임 같은 많은 분야에 적용되어 실용화되고 있다^[1]. 이들 여러 응용분야중 가상현실의 몰입감을 이용하여 교육의 효과를 높이기 위한 몰입형 교육용 시스템 (immersive education systems)의 연구가 많이 수행되었는데, 배의 화재진압 훈련^[2], 우주선의 기계수리^[3] 등의 실제 경험

을 통해 교육을 하기 힘든 분야에서 학습자에게 절차적 지식을 가르치기 위한 연구가 많이 행해졌다. 하지만 이들은 각각 특정분야의 소규모 지식만을 교육하기 위한 시스템이어서 언어교육^[4]과 같은 복잡하고 다양한 지식을 교육하는데는 활용되기 어렵다. 따라서 이를 위해서는 새로운 가상세계를 제공하는 시스템의 설계가 필요한데 본 논문에서는 그 중 가상세계를 구성하는데 바탕이 되는 물리적 측면의 환경 모델링에 중점을 두고자 한다.

가상세계라 함은 컴퓨터로 만들어진 인공적인 세계이고 가상현실의 바탕이 되는 세계이다. 현재까지 가상현실에서 구현된 가상세계는 사용자의 현실감을 위해서 주로 시각, 청각적인 요소를 위주로 설계되었고, 이를 구현하기 위해 데이터장갑이나 입체음향 등의 많은 하드웨어를 사용하였다. 하지만 가상세계에서 느끼는 현실감은 앞에서 말한 감각적 요소뿐만 아니라 가상세계의 주변환경, 에이전트들의 행동, 객체의 움직임 등의 많은 요소들에 의해 영향을 받는다. 특히 사용자가 가상세계에서 바로 접하게 되는 가상환경이 얼마나 실제현실과 유사한가, 즉 현실에서 항상 느끼는 만유인력이나 수시로 변하는 기상현상, 밤낮의 변화 등이 가상세계에 얼마나 적절히 반영되는가는 사용자의 몰입감에 중요한 변수가 될 수 있다. 그 외에도 우리가 현실세계의 공간에서 느낄 수 있는 소리, 빛, 냄새 등의 무형적 요소들 역시 가상환경의 공간 속에 반영되어야 하고, 이렇게 구현된 가상환경과 가상세계 속에 거주하는 에이전트들 사이의 상호작용 방법도 정의하여야 하겠다. 그래서 본 논문에서는 환경을 공간이 가지는 특성이라고 보고, 공간객체를 도입하여 이들을 공간의 기능과 범위에 따라 6개의 계층으로 분류하여 설계한다. 그리고 그 중 최하위 계층에서는 장이론을 이용해서 소리, 빛, 냄새 등의 정보를 파동객체로 모델링하고 에이전트와의 관계도 정의하게 된다.

지금까지 가상세계의 환경은 여러 게임이나 기타 VR 시스템들에서 조금씩 구현이 되어왔다^[1]. 하지만 대부분이 그래픽적인 요소에 치중하거나 또는 단순히 시간의 흐름에 따라 트리거링 되는 변화만을 구현하는데 그치고 있다. 이는 많은 시스템들이 환경보다는 객체나 에이전트의 처리에 중점을 두었기 때문이기도 하지만, 실제와 같은 다양한 환경을 보여주기 위한 환경의 체계적인 구조화 자체가 없었기 때문이다. 따라서 위의 계층화된 환경의 구조화 방안이 지금까지 여러 시스템

의 구현에서 소홀히 취급되었지만, 중요한 역할을 하고 있는 배경 혹은 환경을 좀 더 효율적으로 처리할 수 있는 방법이 될 것이다.

앞으로 2장에서는 관련 연구분야를 살펴본 다음, 3장에서는 구조화된 공간객체를 도입하고 이를 여섯 개의 계층으로 나누어 각각의 기능을 설명하고, 4장에서는 동적인 환경을 구현하기 위한 몇 가지 처리방법을 살펴본 다음, 5장과 6장에서는 시스템의 구현과 결론을 내리고자 한다.

II. 환경 모델링에 관한 관련연구

초기의 가상현실은 HMD 시스템을 이용한 사용자의 몰입도를 증대시키는 연구가 많이 이루어지다 현재는 Immersive VR, Through the window VR, Third person VR의 부분으로 나뉘어서 각각 연구가 진행되고 있다. Immersive VR은 HMD, Data glove 같은 하드웨어 장비를 이용해서 사용자가 실세계와 유사한 감각을 느끼도록 하여 가상세계로의 몰입감을 증대시킨다. 하지만 아직까지도 비싼 하드웨어로 인해 소규모의 응용 분야에만 사용되고 있다. Through the window VR은 그래픽 카드를 이용해서 모니터 상에 화면을 나타내는 방식으로, 간단한 구현방법으로 인해 비행 시뮬레이터, 교육용 시스템 등의 많은 응용분야에 사용되고 있다. 그리고 마지막으로 Third person VR은 사용자가 어떤 하드웨어도 장비하지 않고 비디오 화면과 카메라가 갖추어진 방에 들어가 카메라를 통한 영상의 합성에 의해 가상세계를 경험하게 하는 방식이다. 이는 사용자에게 불편하게 느껴지는 장치들 없이도 쉽게 가상현실을 경험할 수 있는 장점이 있으나 아직은 실험실 수준의 연구가 진행되고 있는 실정이다^[5].

가상세계를 구축하기 위해서 그래픽 분야에서는 많은 연구가 진행되고 있다. 그중 환경을 모델링한 것으로 N. Farenc의 LIG City는 가상도시를 구현하기 위해 정형화된 GIS 데이터베이스를 이용하였다. 즉 도시의 semantic 정보를 이용하여 각각의 ENV(Environment Entities)를 트리형식으로 계층화하고, 각 ENV를 장애물의 역할을 하는 obstacle과 에이전트와의 상호작용을 하는 smart object로 구분하고 이를 바탕으로 에이전트의 도시내의 여러 행동들을 시뮬레이션 하였다. 하지만 LIG City에서 보여주는 환경은 주로 에이전트의 이동을 위한 지리적인 정보만을 제공해 줄뿐 환경자체가

다양한 동적인 환경을 보여주거나 에이전트에 영향을 주지는 못한다^[6].

실세계 환경의 모델링 요소에는 여러 가지가 있지만 그중 인간에게 가장 큰 영향을 미치는 것은 기상현상이라 할 수 있다. 그래서 기상학에서는 날씨를 예측하기 위해 복잡한 방정식을 세워 컴퓨터 시뮬레이션을 하고 있는데 모델링의 지역적 범위에 따라서 GCM, RCM, SEM의 방법을 사용하고 있다. 첫째, GCM (General Circulation Models)은 전지구적인 모형으로써 대기 대순환, 해양의 흐름, 태양의 복사에너지, 구름의 이동 등의 대규모 기후변화를 모델링한다. 둘째, RCM (Regional Climate Models)은 75km정도의 해상도를 가지고 각 지역적인 특성(topology, vegetation...)을 이용하여 GCM과 같이 기후변화를 모델링한다. 셋째, SEM (Spatial Environment Models)은 5km정도의 해상도를 가지고 소규모 지역별 특성을 적용하여 매일의 일기변화를 예측하는데 사용이 되고 있다. 과거에 이들 각각의 모델들은 연구분야에 따라 선택적으로 사용되어 왔으나 각 모델들이 실제적으로는 독립적이지 않고 상호 보완적이기 때문에 최근에는 이들 각 모델들을 통합하여 좀더 정확한 날씨를 예측하려는 시도가 있다^[7].

앞에서 언급한 연구 이외에도 가상세계의 구현에 관련된 시스템들은 많이 있다. 하지만 이들은 대개 환경을 수동적인 배경만으로 처리하였거나^[4,8], 특수한 목적을 위해 설계하였다^[9]. 그리고 많은 경우 ad hoc 디자인을 함으로써 범용적으로 가상세계에 적용하기는 어려웠다. 이는 많은 VR 시스템들^[10,11]이 환경의 논리적 처리보다는 사용자 인터페이스나 시각적인 그래픽 렌더링에 치중한 결과이다. 따라서 본 논문에서는 VR 시스템의 설계시 환경의 외부적인 요소와 더불어 내부의 논리적 구조설계에 중점을 둘 것이고 이를 위해 위에서 언급한 기상학이나 물리학 등의 연구결과를 참조하여 실세계와 유사하면서 범용적이고 일반화된 환경의 구현에 초점을 맞출 것이다.

III. 환경의 구조

에이전트는 자신을 둘러싸고 있는 주위환경으로부터 시각, 청각적인 정보를 얻고 그것을 바탕으로 판단하여 반응함으로써 경험과 학습을 하게 된다. 가상세계에서 에이전트를 둘러싸고 있는 환경으로는 집, 마을, 지구, 우주 등 여러 가지가 있을 수 있지만, 에이전트의 관점

에서는 자신을 둘러싸고 있는 공간만을 주위환경으로 인식하게 된다. 즉 에이전트가 집안에 있을 때는 집이란 환경을 산에 있을 때는 산이라는 환경만을 인식한다. 따라서 전체 환경 설계시 에이전트의 주위환경 이외의 환경은 내부시스템에서 처리하고 각 외부환경은 내부환경의 영향을 받아 동작하도록 구성할 수 있다. 그림 1은 가상세계를 에이전트와 환경의 관점에서 도식화 한 것이다. 여기서 에이전트는 4개의 환경에 속해 있고 이들 환경으로부터 영향을 받고 있다. 이것을 식으로 나타내면 $Agent \in House$ 그리고 $House \subset Village \subset Earth \subset Universe$ 되는데, 이것은 다시 $Agent \in House, Village, Earth, Universe$ 로 나타낼 수 있다. 따라서 에이전트가 집안에 있다는 사실만을 가지고도 전체 가상세계를 구성하고 동작시킬 수 있다. 각 환경계층들 사이의 화살표는 상호간의 영향의 가능성을 나타낸다.

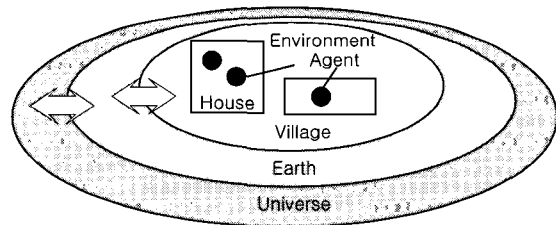


그림 1. 환경과 에이전트
Fig. 1. Environment and agent.

1. 환경의 구성요소

가상세계에서 동적인 환경을 구현하기 위해서는 우선 현실세계에 대한 관찰이 필요하다. 이는 여러 학문적인 연구가 필요한 부분이지만, 여기서는 인간에게 직접적인 영향을 주는 환경요소들을 컴퓨터 시뮬레이션 측면에서 살펴보겠다. 일반적으로 우주는 시간, 공간으로 구성된 환경과 그 안의 객체들로 구성되어있다. 여기서 시간은 일차원의 선형적 흐름으로 모델링하고 공간은 삼차원으로 모델링을 할 수 있다. 그리고 객체는 스스로의 판단에 의해 행동할 수 있는 에이전트와 그렇지 않은 일반객체로 구분할 수 있다. 이들 객체는 공간 안에 위치하여 공간에 의해 영향을 받거나 공간의 환경을 변화시키면서 존재한다. 공간을 구성하는 환경적 요소로는 기온, 밝기, 바람, 소리, 전자기장등 정의하기에 따라서는 수많은 요소들을 열거 할 수 있을 것이다. 하지만 많은 환경요소중 인간에게 광범위한 영향을

주는 것은 대개 날씨, 만유인력 등의 몇 가지 자연현상에 국한됨을 알 수 있다. 따라서 가상환경을 구축할 때 이들 몇몇 환경변수들을 구현함으로써 사용자나 에이전트가 가상세계를 실제 현실과 비슷하게 느끼게 되고 따라서 가상현실로의 몰입감을 얻게 될 것이다.

본 논문에서는 실세계 환경중 지구환경을 이루는 요소변수들을 각 변수가 발생하는 원리에 따라 크게 객체특성, 동적 특성, 국지적 특성의 3가지 종류로 나누었다. 첫째, 객체 특성은 지구라는 질량을 가진 물질이 존재함으로써 인해 생기는 특성으로 중력, 지자기, 대기권 등을 들 수 있고, 둘째, 동적 특성은 지구의 자전과 공전으로 인해 발생하는 특성으로 밤낮의 변화, 기온의 일 변화 연 변화 등을 들 수 있다. 마지막으로 국지적 특성은 지구 표면의 국지적인 지형 지물이나 동식물 생태계에 의해 발생할 수 있는 비, 눈, 바람, 안개 등의 현상을 말한다. 이와 같이 분류된 각 특성들은 아래에서 정의하는 공간객체의 각각의 계층에서 구현이 될 것이다.

2. 공간객체

공간객체란 에이전트를 둘러싸고 있는 환경을 객체 지향 기법으로 나타낸 것이다^[12]. 우주공간을 제외한 대부분의 공간객체는 물리적 객체에서 유도되어 나오는 유도 공간객체이다. 물리적 객체가 하나 생성되면 그 객체의 외부공간과 내부공간이 생긴다. 인간이 살고 있는 환경은 지구객체에 의해 생성된 외부공간이고, 우리들이 거주하는 집은 집이란 건물객체에 의해 생성된 내부공간이다. 특이한 경우로 도로, 인도 같은 지형물에 의해 생긴 공간도 외부공간이라 할 수 있다. 이들 유도 공간객체는 각각의 객체 특성에 따라 다른 환경을 이루고, 서로 공간상의 포함관계에 따라 계층구조를 이루으로써 에이전트의 이동에 효과적으로 이용될 수 있다^[6]. 하지만 유도 공간객체는 각 물리적 객체마다 생성되므로 각각이 공간에 대한 환경정보를 가진다는 것은 비효율적이고 또한 지구와 같은 거대한 환경을 하나의 공간객체로 취급함으로써 실제의 다양한 상황을 구현하기가 힘들다는 단점이 있다. 따라서 이 문제를 해결하기 위해 특성 공간객체를 도입한다. 특성 공간객체는 유도 공간객체를 공간의 물리적 특성(온도, 습도, 강우...)의 차이에 따라 여러 개로 구분 짓는데 예를 들어 지구 공간객체는 산, 강, 사막, 초원, 바다 등의 다른 특성을 지닌 여러 특성 공간객체로 구분할 수 있다. 이들

특성 공간객체에는 환경을 구현하기 위해 각 공간객체 별로 다른 변수들을 가지고 기상현상과 같은 물리적 현상을 구현하게 된다.

가상환경 시스템(VENV)의 전체적인 구조는 그림 2와 같이 6개의 Layer로 구현이 된다. 각 Layer는 시간의 흐름에 따라 독립적으로 동작하면서 인접 Layer과 정보를 교환하여 서로 유기적으로 결합하게 된다. 이를 위해 공간객체(S)는 다음과 같이 정의된다.

$$S = (Oid, Oid_h, Oid_l, Area, SpaceType, Layer, Char, Env, Processor)$$

여기서 Oid는 각 공간객체의 식별자가 되고, Oid_h와 Oid_l는 그 공간객체의 인접 Layer의 상위 공간객체와 하위 공간객체의 식별자가 된다. Area는 각 공간객체가 차지하는 범위를 나타내고, SpaceType은 공간객체가 특성 공간객체인지 유도 공간객체인지를 나타낸다. Char와 Env는 각 공간객체가 가지는 특성변수와 환경변수들을 나타낸 것이고, Layer는 각 공간객체가 속하게 되는 계층을 나타낸다. 여기서 특히 환경변수는 그림 3에서 보는 바와 같이 현재 Layer의 특성변수와 상위 Layer의 환경변수를 조합하여 계산한다. 따라서 $Env = Processor(Char, Oid_h.Env)$ 로 정의된다. 추가로 각 Layer간의 공간객체는 다음과 같은 관계가 있다.

- 하위 Layer의 공간객체는 상위 Layer의 공간객체에 포함된다. ($S_{L_{N-1}} \supset S_{L_N} \supset S_{L_{N+1}}$)
- 하위 Layer의 공간객체들을 전부 합하면 상위 Layer의 공간객체와 같다. ($S_{L_N} = \sum_{i=0}^m S_{L_{N+1}}^i$)
- 상위 Layer의 공간객체는 주기적으로 하위 공간객체에 정보를 전달한다.

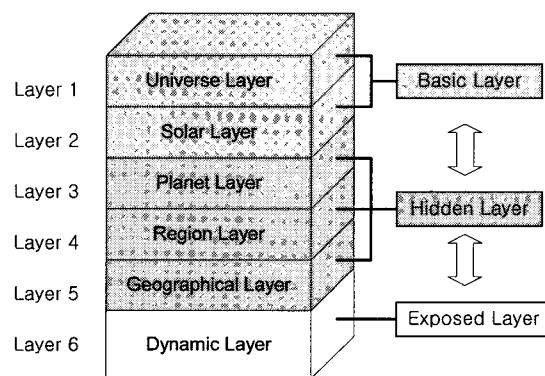


그림 2. 공간객체의 계층구조
Fig. 2. Layered architecture of space objects.

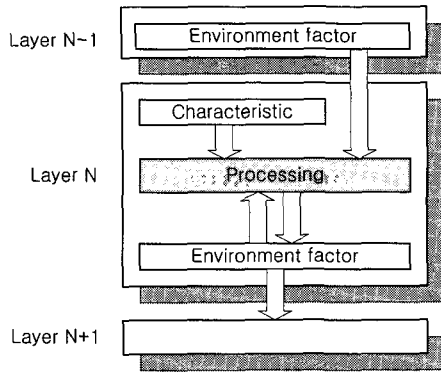


그림 3. 공간객체 사이의 상관관계
Fig. 3. Relation between space objects.

IV. 환경의 동작

1. 시스템의 동작

VENV는 각 계층의 기능에 따라 Basic Layer, Hidden Layer, Exposed Layer의 3개의 Layer로 나누어진다. 우선 Basic Layer는 Universe Layer와 Solar Layer로 이루어진 것으로 가상세계의 환경을 이루는 기본 계층이다. 이들 계층은 실제 가상세계의 구현에 직접적인 영향을 주기보다는 우주공간을 구성하는 기본 정보만을 가진다. Hidden Layer는 Planet, Region, Geographical Layer로 나누어지는데 가상환경에 드러나지는 않지만 내부에서 가상환경의 기상현상이나 자연현상을 구현하는 역할을 수행한다. Exposed Layer는 가상현실에서 직접 구현된 공간으로 이루어지는데 여기서는 에이전트와의 상호작용을 위한 여러 가지 역할을 수행한다. 각각의 세부 Layer에 대해서 자세히 살펴보면 다음과 같다.

Universe Layer(UL)는 우주공간을 지칭하는 계층으로 특별한 환경변수 없이 공간객체의 Meta Object의 역할만을 수행하는 최상위 계층이다. Solar Layer(SL)는 항성계를 나타내는 것으로 일반적으로 태양계로 구현된다. SL에는 항성계를 이루기 위한 항성의 무게, 밝기, 에너지 등 천체 물리학적 정보를 바탕으로 항성계의 구조와 행성의 운동을 정의한다.

Planet Layer(PL)는 행성객체에 의해 생성되는 유도 공간객체로 이루어진 계층이다. PL에서는 행성의 질량, 자전주기, 공전주기, 항성까지의 거리 등의 특성과 SL에서의 항성에너지, 밝기 등의 정보를 바탕으로 행성의 중력, 평균기온, 밝기 등의 전체적인 행성환경을 계산한

다. Region Layer(RL)는 행성에 의해 생성된 유도 공간객체를 위도, 경도를 기준으로 다시 구분한 특성 공간객체로 구성된다. 보통 행성은 위도에 따라 각기 다른 환경을 구성하는데, 특히 지구 같은 경우 지축의 경사로 인해 1년 주기로 위도에 따라 4계절의 변화를 보이게 된다. 따라서 RL에서는 특성변수인 위도, 경도, 지축경사를 바탕으로 가상현실에서 구현된 지역에서의 기온, 바람, 밝기, 강우 등의 정보를 계산한다. Geographical Layer(GL)는 RL의 공간객체를 산, 초원, 사막, 평야와 같은 지형적인 특성에 따라 분류한 특성 공간객체로 구성된다. 여기서는 각 지형의 타입에 따라서 같은 지역이라도 다른 환경을 이루게 된다.

Dynamic Layer(DL)는 최하위 계층으로 가상현실 상에서 구축된 인공 혹은 자연 구조물에 의해 생긴 공간객체로 이루어진다. DL에서는 GL의 환경변수를 받아서 그대로 사용하거나 구조물의 특성에 따라 환경변수를 조금씩 변화시킬 수 있다. 그리고 각 구조물은 공간의 포함관계와 위치에 따라 Tree형식으로 데이터베이스에 저장되어 추후에 에이전트의 이동에 이용된다. 그림 4는 각 Layer의 Char, Env 변수를 정의한 것이다.

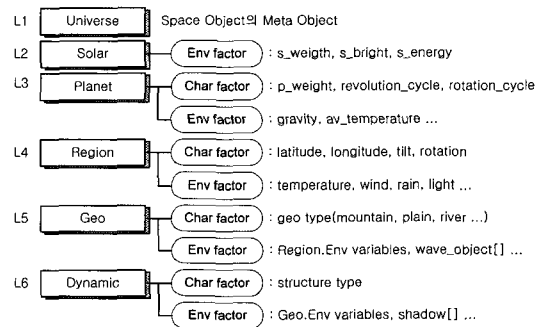


그림 4. VENV 시스템에서 변수 정의
Fig. 4. Definition of variables for the VENV system.

2. 환경변수의 구현

환경변수중 첫째, 객체특성은 각 객체의 생성시 발생하는 유도 공간객체 안에 변수로 나타낸다. 예를 들어 지구 객체를 생성하면 지구의 대기, 중력 등의 변수를 지구 공간객체에 설정함으로써 간단히 나타낼 수 있다. 둘째, 동적특성은 Hidden Layer인 3~4층의 PL, RL에서 구현이 되는데, 이 특성을 구현하기 위해서는 물리학, 기상학적인 많은 지식들과 계산이 요구된다. 하지만 이들 값이 보통 1년 혹은 하루를 주기로 계속 반복된

다는 점에 착안해 주기함수($f(t)=A \cdot \sin(t/p)$)를 사용하여 연 변화, 일 변화를 모델링한다. 예를 들어 기온의 일변화를 나타내는 주기함수의 경우 아침은 $|f(t)|=0+$ 으로, 저녁은 $|f(t)|=0-$ 로, 정오는 $|f(t)|=A$, 자정은 $|f(t)|=-A$, 단 $|f(t)|$ 는 $f(t)$ 의 순간 값 그리고 A 는 기온의 최대 절대값 등으로 구현을 하고, 전체 연 기온은 일 변화를 처리하는 주기함수와 연 변화를 처리하는 주기함수 두 개의 합으로 구성한다. 그림 5는 중위도 지방의 기온과 밝기의 변화를 시뮬레이션 한 결과이다. 그림에서 상단의 그래프는 기온과 밝기의 일 변화를 나타내었고, 중간 그래프는 일일 평균기온과 일일 평균밝기를 연 변화로 나타내었다. 여기서 기온의 일변화의 경우 계절에 따른 일교차의 변화를 포함하였다. 최종적으로 하단의 그래프는 위 두 그래프를 더함으로써 만들어진다. 각각의 환경변수에 대한 자세한 계산과정은 다음과 같다.

$$\text{Region. temperature} = \alpha_t \times \sin(t) + (\beta_t + k_t \times \cos(t)) \sin(r \times t) \pm \delta_t$$

$$\text{Region. light} = \alpha_l \times \sin(t) + k_l \times \sin(r \times t) \pm \delta_l$$

where $\alpha, \beta = f(\text{latitude, tilt, rotation})$, $\delta = g(t)$:random function

$r = \text{revolution_cycle}$, $t = \text{time}$, $k_i = \text{coefficients}$

Hidden Layer에서 구현되는 환경변수 사이에는 비가

오면 온도가 내려가고 습도는 올라가는 것과 같이 서로간에 밀접한 인과관계가 존재한다. 이들 인과관계를 전부 추적하여 구현할 수는 없지만 본 시스템에서는 기본적인 몇 개의 변수들간의 인과관계를 규칙 기반으로 해서 각 공간객체에 함수로 구현을 한다. 마지막으로 여기서 계산된 환경변수는 주기적으로 하위 계층의 공간객체에 전달되어 하위 공간객체의 환경변수가 갱신되도록 한다.

셋째, 국지적 특성은 실제 가상세계를 구성할 때 배경이 되는 공간의 특성을 나타낸 것으로, 가상현실의 맵을 제작할 때 각 지형별로 특성변수를 설정함으로써 구현이 가능하다. 그리고 또 국지적 특성은 각 지형별 특성이외에도 한 지역내의 인접 공간객체에 의해서도 영향을 받게 된다. 일반적으로 한 지역은 여러 지형을 가진 공간객체의 격자형태로 구성되어 있다. 그림 6은 인접한 공간객체가 정보를 어떻게 처리하는가를 보여 준다. 우선 각 셀 마다 가중치가 매겨진 3×3의 마스크(Mask)를 이용한다. 마스크의 선택은 그 지역의 대기의 흐름, 바람의 방향에 따라 base point(bp) 위치와 각 셀의 가중치가 다른 마스크를 적용할 수 있는데, 여기서는 편서풍이 부는 중위도 지방을 나타낸 것으로 첫 번째 마스크를 적용하였다. 그리고 아래의 5×5의 셀은 각 지역별 기온을 보여주고 있는데 현재 에이전트가 위치하고 있는 (3,3)공간의 기온은 21℃이고 그 서쪽지형(3,2)의 기온은 25℃로 4℃가 더 높게 형성되어 있다.

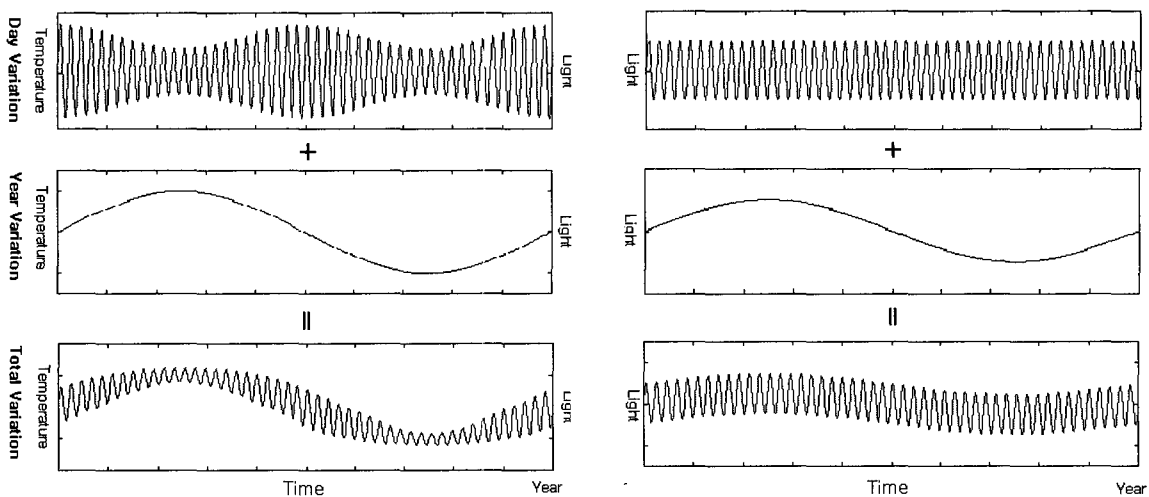


그림 5. 기온 및 밝기의 변화 모델링

Fig. 5. Modelling of variations of temperature and light.

여기서 아래의 식을 이용하여 마스크를 적용하면 다음 순간의 기온분포를 구할 수 있다.

$$Temperature(m, n) = \frac{1}{25} \sum_{i=1}^3 \sum_{j=1}^3 Mask(i, j) \times Temperature(m - bp.x + i, n - bp.y + j)$$

계산된 결과, 아래의 오른쪽 셀을 살펴보면 (3,3)지역은 서쪽의 영향을 받아서 온도가 23℃로 낮아졌음을 볼 수 있다. 여기서는 기온에 대해서만 언급하였지만 기타 환경변수들도 적절한 수치화 모델만 가능하면 이 방법의 적용이 가능할 것이다. 그리고 마스크도 지역의 크기나 대기의 흐름의 세기에 따라 5×5나 그 이상으로 확장하여 적용할 수 있을 것이다.

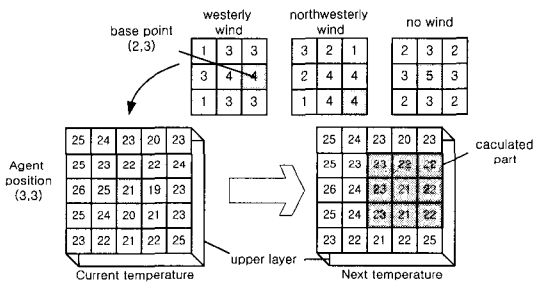


그림 6. GL 공간객체들 사이의 상관관계
Fig. 6. Relations among GL space objects.

3. 공간객체내의 파동객체 구현

가상현실의 장면에는 빛, 소리, 냄새 등의 발원지를 가지고 주변으로 퍼져나가는 환경변수들이 존재하는데 이들을 파동객체라 하고 다음과 같이 정의한다.

Wave Object=(type, source, level, duration, contents)

여기서 type은 파동객체의 종류를 나타내고, source는 공간상의 발원지 좌표, level은 각 파동객체들의 세기, duration은 지속시간, contents는 파동객체의 종류를 나타낸다. 이렇게 정의된 파동객체는 각 공간객체 상에서 배열의 형태로 저장되어 에이전트가 인식을 하게 된다.

그림 7은 가상세계에 에이전트와 파동객체가 등장시 발생하는 인식장 및 영향장을 도식화 한 것이다. 여기서 각 에이전트는 자신이 인식할 수 있는 범위를 나타내는 인식장(Sensing Field)을 가지게 되고, 파동객체는 자신이 영향을 줄 수 있는 영역을 나타내는 영향장

(Stimulus Field)을 가지게 된다. 그래서 각 객체들이 서로의 영역에 포함이 될 때, 즉 (Agent ∈ Stimulus Field) AND (Wave Object ∈ Sensing Field)가 될 때 에이전트는 파동객체를 인식하게 된다. 그림 7의 (b)는 파동객체의 영향장이 작아서 에이전트를 포함하지 못함으로 인해 에이전트가 인식을 하지 못하는 경우를 나타낸 것이고, (c)는 그 반대로 에이전트의 인식장이 좁아서 파동객체를 인식하지 못하는 경우를 나타낸 것이다.

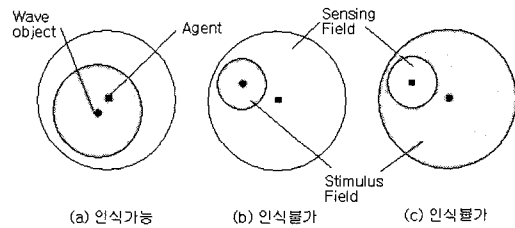


그림 7. Agent의 파동객체 인식 모델
Fig. 7. Modelling of agent's recognition of wave object.

가상세계에 파동객체가 생성시 이들 파동객체는 에이전트의 위치에서의 물리적 세기에 따라 정렬이 이루어지고 에이전트는 이를 참조하여 세기가 큰 것을 우선 인식하게 된다. 전체적인 파동객체의 인식 알고리즘은 다음과 같다.

```

Algorithm /* Recognize Wave Object */
For All wave_object
  IF wave_object.type = agent.sensing_type AND
  distance(wave_object.agent) < wave_object.stimulus_field AND
  distance(wave_object.agent) < agent.sensing_field
  /* If wave object can be recognized by agent */
  Calculate level at agent's location
  Add wave_object to temp
Sort temp with level
Recognition Top n temp
    
```

파동객체는 그 특성상 영향장 안에 어떤 장애물이 있을 때 그 장애물 뒤로는 영향장이 형성되지 못하는 그림자 부분이 발생한다. 따라서 각 공간객체에는 이 그림자를 처리하기 위해 따로 리스트를 두어 공간의 어느 부분이 그림자에 해당하는지를 나타낸다. 그림자는 다음과 같이 정의한다.

Shadow = (type, location, level)

여기서 type은 파동객체의 종류를 나타내고, location은

공간상의 위치, level은 그림자에서의 파동객체의 세기를 나타낸다.

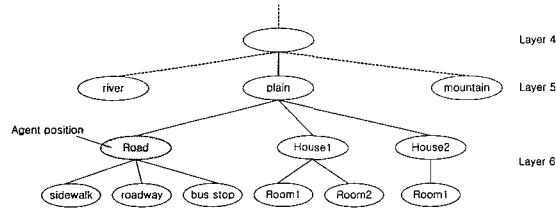


그림 8. DL 공간객체의 Tree구조
Fig. 8. Tree structure of DL space objects.

4. Dynamic Layer의 구현

최하위 계층인 DL은 에이전트와 직접 접하게 되는 계층으로 이들과의 인터페이스 처리가 중요한 기능이 된다. 각 에이전트는 여러 공간객체중 어느 하나에 속하게 되고 그 공간객체의 환경에 영향을 받는다. 그림 8에서 보면 에이전트는 도로에 위치하고 있는데 도로의 환경은 5번째 계층의 plain 공간객체의 환경에 영향을 받게된다. 그래서 만약 plain공간의 기온이 내려가면 도로 역시 기온이 내려간다. 그러면 에이전트는 DL 공간객체의 Tree구조를 이용하여 House1으로 이동한다. House1의 현재 환경은 역시 plain 공간과 같은 상태이므로 에이전트가 들어오면 추위를 느끼고 난로를 피운다. 그러면 House1의 온도는 상승하게 되고 이는 다시 에이전트에게 영향을 주어 에이전트는 만족을 느끼게 된다. 이와 같이 에이전트는 자신이 감지하고자 하는 시각, 청각, 촉각 등의 정보를 자신이 거주하고 있는 공간에 접근함으로써 얻을 수 있다. 나머지 필요한 정보

들은 주기적으로 공간객체들 사이에 처리가 되어 갱신이 이루어짐으로써 다양한 상황 속의 에이전트의 행동 구현이 가능하다.

V. 구현

위에서 제시한 환경구조를 바탕으로 실세계와 유사한 다양한 가상환경을 구현할 수 있을 것이다. 완전한 가상환경을 구축하기 위해서는 여러 환경요소들의 모델링과 이를 시뮬레이션 하기 위한 시스템이 필요하다. 하지만 환경요소간의 인과관계로 인한 복잡성 그리고 현재 시스템의 성능의 한계 등으로 인해 구현에 어려움이 따른다. 그래서 여기서는 대표적인 환경변수인 밤낮의 변화, 기온, 바람 등의 몇몇 요소를 구현하고, 이들과 에이전트와의 상호작용을 보임으로써 그 가능성을 보였다. 시스템은 Desktop VR방식으로 Window환경에서 구현을 하였고, Visual C++와 3차원 그래픽 처리를 위해 Morfit 3차원 라이브러리를 사용하였다. 그림 9는 중위도 지방의 마을을 가상세계로 구현한 것이다.

왼쪽의 상태 창에는 현재 가상세계의 시간과 기온, 빛, 바람 등의 환경변수, 환경과 에이전트의 상황묘사를 위한 창이 표시되어 있고, 오른쪽은 마을과 에이전트의 상황을 3차원으로 보여주고 있다. 구현된 시스템의 가상환경은 앞에서 제시한 6개의 계층을 각각 클래스로 구성하였다. PL의 공간객체는 기온, 빛의 연 변화 정보를 처리하고, RL에서는 각 환경요소들의 일 변화 정보를 주기함수로 처리하였다. GL에서는 바람의 방향과

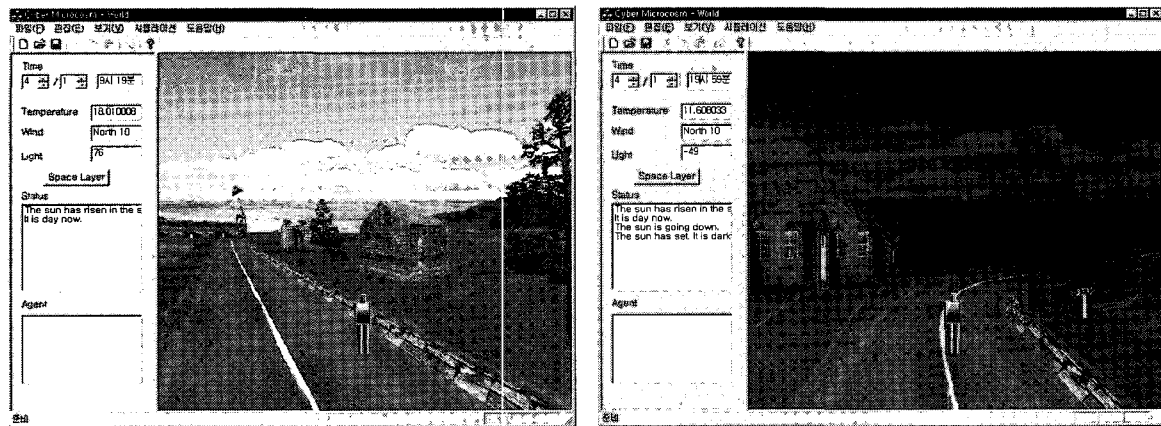


그림 9. 구현된 가상세계의 낮과 밤 환경
Fig. 9. Typical daytime and nighttime in the virtual world in implementation.

주변지형에 의한 기온의 변화를 앞에서 제시한 국지적 특성의 처리방법을 이용하여 계산하였다. 그리고 가상 세계에서의 동물의 소리와 전등 빛에 대한 정보를 DL에서 파동객체로 처리하였다. 마을과 에이전트는 Editor를 통해 설계를 하고 그래픽 라이브러리에 의해 렌더링된다. 환경변수들은 1분단위로 갱신이 이루어지고 이것은 다시 전체 시스템에 반영되어 3차원으로 렌더링되어 보여진다. 에이전트는 이들 환경변수를 주기적으로 감지하여 판단하고 행동함으로써 환경에 적응하게 된다. 예를 들어 시스템 상에서 에이전트는 시간의 흐름을 내부 타이머가 아니라 공간객체의 빛이란 변수를 감지하여 변수값이 +인지 -인지, 크기가 어느 정도인지를 판단하여 현재의 시간 또는 환경을 인식하게 된다.

구현된 가상세계에서 도로, 나무, 집 등은 여러개의 다각형(polygon)을 가진 그래픽 객체로 구성하고, 에이전트는 다양한 행동을 위해 팔, 다리, 머리 등의 13부분을 각각 그래픽 객체로 처리하였다. 그리고 도로, 집과 같은 지형지물의 경우에는 유도공간객체를 추가로 생성하고, 이 유도공간객체는 DL에서 Tree구조로 저장되어 에이전트의 이동에 이용되게 된다^[12]. 그림 10은 새벽과 저녁의 가상환경에서의 에이전트의 행동을 보여주고 있다. 그림의 좌측 화면은 새벽이 되어 에이전트가 조깅하는 상황을 보여주고 있고 우측의 화면은 에이전트가 양의 울음소리를 듣고 있는 장면을 나타내고 있다. 여기서 우측의 에이전트는 DL의 공간객체내의 파동객체를 검색하여 제일 근접한 곳에서 들이는 양의 울음소리를 인식하게 된다.

최종적으로, 가상환경의 구조화 방안을 바탕으로 설

계된 위의 시스템은 시간과 위치에 따라 현실과 유사한 다양한 상황을 보여주었다. 이는 이전의 많은 가상 현실 시스템에서 단순한 배경으로만 사용된 환경을 동적으로 다양하게 변하고, 의미 있는 정보를 가진 환경으로 바꾸었다. 구현된 시스템에서 사용자들은 현실세계와 유사한 상황을 연출하는 환경에 의해 더욱더 몰입감을 느끼는 것을 실험을 통해 확인할 수 있었다. 이는 인간이 단순한 감각에 의해서만이 아니라 지금까지의 자신의 경험이나 지식을 통해 상황을 판단하고 몰입한다는 것을 의미한다. 따라서 가상세계 구축시 환경과 여러 물리적, 논리적 지식들의 내부적인 구조설계가 중요하고 필요하다 하겠다.

VI. 결론 및 향후 연구방향

본 논문에서는 가상세계를 구축함에 있어서 기본이 되는 주위 환경의 구조화를 통해 실세계와 유사한 다양한 환경의 구축을 시도하였다. 우선 실세계의 환경을 구성하는 요소변수들을 분류, 정리하고, 각 요소들을 구현하기 위해 공간객체를 도입하여 그들을 6개의 계층으로 구분하였다. 그리고 6개의 계층은 기능에 따라서 자연현상을 처리하는 Hidden Layer와 에이전트와 상호작용하는 Exposed Layer로 구분하여 효율적인 처리방법을 제시하였다. 그리고 각 Layer에서 환경요소들을 구현함에 있어서 주기적인 특성을 가진 환경 변수는 주기함수를 이용하여 간단하면서도 효과적인 모델링을 시도하였으며, 마지막으로 Dynamic Layer에서는 빛, 소리, 냄새 같은 발원지를 가진 환경변수를 파동객체로

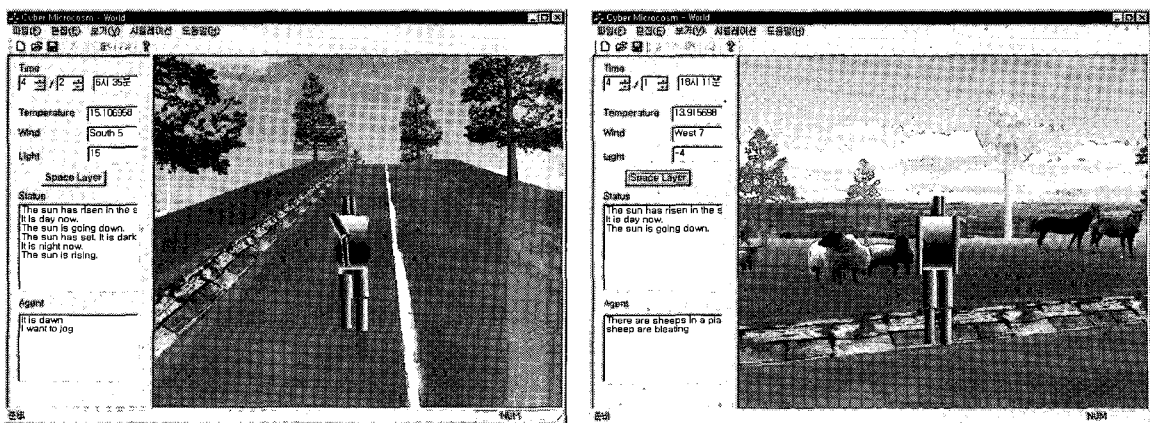


그림 10. 가상환경 속의 에이전트
Fig. 10. Agent in virtual environment.

정의하여 처리하였다.

여기서 제시한 가상환경의 계층화된 구조화 방법과 각 환경요소의 모델링 방법들을 사용하여 구축한 가상 세계는 다음과 같은 몇 가지 장점을 가진다. 첫째, 환경을 독립적인 6개의 Layer로 구성함으로써 다양성의 극대화를 이룰 수 있다. 이는 마치 여러 에이전트에 의해 이루어진 시스템이 이들 각각의 행동에 의해 다양한 상황을 연출하는 것과 비슷하다. 그리고 이 다양성은 환경에 그치는 것이 아니라 그 속에 있는 에이전트에 영향을 주어 다양한 행동을 유발하는 역할도 하게 된다. 둘째, 가상환경에서 지리, 기상, 소리, 빛 등의 정보를 처리하고 에이전트와의 간단한 인터페이스를 유지함으로써 에이전트의 설계, 구현이 단순해진다. 이는 복잡한 에이전트의 설계시 내부적인 정보처리 과정에만 집중할 수 있도록 한다. 마지막으로, 유연한 확장성을 가진다. 여기서 제안된 환경은 계층구조를 기본구조로 하고 있어서 필요에 따라 변수나 기능을 추가함으로써 필요한 가상시스템을 효과적으로 구현할 수 있다.

앞으로의 연구방향은 특성 공간객체의 정의시 생기는 경계문제를 명확히 처리하고, 또 이들 공간객체들이 현재 하나의 시스템에서 처리되는 것을 네트워크를 통한 분산 처리가 가능하도록 하여야 할 것이다. 그리고 현재는 물리적 환경의 관점에서만 모델링을 하였지만, 추후에 에이전트가 많이 등장할 때 유사한 방법으로 에이전트의 심리적 환경 혹은 문화적 환경으로의 확대 모델링이 가능할 것이다.

참 고 문 헌

- [1] J. Nomura, "Virtual reality technology and its industrial applications", IEEE Control Engineering Practice, Vol.7, pp.1381-1394, 1999.
- [2] D. Tate, L. Sibert, and T. King, "Using Virtual Environments for Firefighter Training", IEEE Computer Graphics and Applications, Vol.17, No.6, Nov., pp.23-29, 1997.
- [3] R. Loftin and P. Kenny, "Training the Hubble Space Telescope Flight Team", IEEE Computer Graphic ana Applications, 15(5), pp.31-37, 1995.
- [4] E. Ohmaye, "Simulation-based language learning", Northwestern Univ., Ph.D dissertation, 1992.
- [5] J. Raper, "Georeferenced four-dimensional virtual environments: principles and applications", IEEE Computer, Environ. and Urban Systems, Vol.22, No.6, pp529-539, 1998.
- [6] N. Farenc, "An Informed Environment dedicated to the simulation of virtual humans in urban context", EUROGRAPHICS'99, Vol.18, No.3, 1999.
- [7] R. Lau and A. Young, "Downscaling global information for regional benefit: coupling spatial models at varying space and times scales", IEEE Environmental Modelling & Software, Vol.14, pp.519-529, 1999.
- [8] J. Murray and S. Malone, "The structures of advanced multimedia learning environments: reconfiguring space, time, story, and text," ITS '92, Montreal, Canada, June 1992.
- [9] A. Bimbo, "Specification by-example of virtual agent behavior," IEEE Visualization and Computer Graphics, Vol 1, No 4, pp.350-360, 1995.
- [10] F. Neyret, "Modelling, animating, and rendering complex scenes using volumetric textures," IEEE Trans. on Visualization & Computer Graphics, Vol.4, No.1, pp.55-70, 1998.
- [11] H. Noser and D. Talmann, "A Rule-Based Interactive Behavioral Animation System for Humanoids", IEEE Trans. Visualization and Computer Graphics, Vol.5, No.4, pp.281-307, 1999.
- [12] 박종희, 박형근, "동적 환경에서의 효율적인 자율 에이전트 행위의 설계 및 구현", 정보과학회 추계 학술대회, Vol 26, No 2, pp.251-253, 1999

저 자 소 개



金 魯 淳(正會員)

1973년생. 1999년 2월 경북대학교
전자공학과(학사), 2001년 2월 경북
대학교 전자공학과(석사), 2001-현
재 LG전자 차세대 단말연구소 연구
원. 주관심분야는 인공지능, 가상현
실, IMT-2000

朴 宗 熹(正會員) 第 32 卷 B 編 第 12 號 參照