

다양성과 재사용성을 가진 가상 인간의 기본 행동 구현

Implementation of the basic Actions for Virtual Human with Diversity and Reusability

김유신, 정근재, 박종희
경북대학교 IT대학 전자공학부

Yu-Shin Kim(isbn99@ee.knu.ac.kr), Geun-Jae Jung(cooky8884@ee.knu.ac.kr),
Jong-Hee Park(jhpark56@ee.knu.ac.kr)

요약

본 논문은 가상세계속의 가상인간의 다양한 행동들을 제한된 개수의 기본행동들로써 구현하고자 한다. 먼저 객체들의 속성과 행동 등을 통합적으로 나타내주는 지식베이스인 온톨로지를 사용하여 다양한 행동들을 행동계층구조로 조직하고 그 속에서 행동들은 세부 행동들로 구체화한다. 가상인간의 기본행동은 기초 동작들의 조합으로 만들어지고 기본행동들을 최대한 재사용함으로써 다양한 행동들을 효과적으로 구현할 수 있게 설계한다. 이 때 기본행동들을 정밀성이나 사실성보다 간소함과 다양성에 초점을 맞추어 구현하고자 한다. 이를 바탕으로 가상 에이전트들의 특성을 파악하고 특성에 따라 에이전트가 고유의 패턴을 가지고 동작을 생성하는 방법을 개발한다. 또한 가상의 에이전트가 주변의 정성적 변화에 적응하여 동작을 변화하는 방법을 제시한다. 이렇게 제시된 방법들을 인간의 실제 움직임들 즉, 걷기, 달리기, 던지기 등의 동작에 적용해 실현가능성을 검증한다.

■ 중심어 : | 인간 행동 | 기초 동작 | 행동계층구조 |

Abstract

In this paper, we attempt to realize the diversity of human actions in a virtual world in terms of the basic actions. We utilize the comprehensive knowledge structure of ontology to organize the human actions into an action hierarchy, each action being the root action of its associated hierarchy of specialized actions. Each basic action is implemented by composing the primitive motions and each basic action in turn by reusing those basic actions. Our approach to the development of these actions is focused more on their diversity and brevity than on their visual realism. Based on this design we develop a method to grasp virtual humans' characteristics and reflect those characteristics on their respective action patterns. We also present a mechanism for the agents to adapt their motions to diverse qualitative changes in environment. We apply these methods to such basic actions as walk, run and throw to demonstrate their viability.

■ keyword : | Human Action | Primitive Action | Action Hierarchy |

* 이 논문은 BK21 경북대학교 인력양성 사업단과 경북대학교 산학협력단의 지원을 받아 연구 하였습니다.

접수번호 : #110408-006

심사완료일 : 2011년 05월 09일

접수일자 : 2011년 04월 08일

교신저자 : 정근재, e-mail : cooky8884@ee.knu.ac.kr

I. 서론

현실감 있는 가상세계를 구축하기 위한 중요한 고려 사항 중의 하나는 사용자들에게 지속적인 흥미와 관심을 유발하는 것이다. 이를 위해서는 가상현실 내에 일어나는 자연현상들과 에이전트들의 움직임이 사람들에게 다양성과 신뢰성 그리고 자연스러움을 줄 수 있어야 한다[1]. 이러한 목적의 일환으로 최근 3차원 게임이나 휴머노이드 로봇과 같이 사람의 동작을 사실적으로 묘사하기 위한 노력이 활발히 이루어지고 있다[2][3]. 그리고 컴퓨터 애니메이션 분야에서도 아이콘 이용 방식이나 모션 캡처 방식 등 다양한 기법들을 이용하여 움직임을 표현함에 있어서 역학적인 안정성과 실물과의 유사성을 위한 활발한 연구가 진행되고 있다[4][5]. 하지만 이러한 기존의 연구들은 동작의 사실성(realism)에 치중하여 다양한 에이전트들의 각기 다른 동작 패턴과 환경 변화에 대응한 움직임의 변화를 표현하는 데에는 한계가 있고, 가능한 동작의 종류가 제한되어 다양한 동작 행태의 사실적 표현에는 적합하지 않다. 또한 에이전트의 예기치 못한 행위에 대해 대처하기 어렵고, 배경이 되는 객체들과 사이의 상호작용을 수행할 방법을 제공하지 못한다.

본 논문에서는 이러한 기존의 연구가 초점을 맞추고 있는 동작의 사실성보다는 다양성과 효율성을 지향하는 동작 모델링 기법을 제안하고, 정교하지만 제한된 동작을 할 수 있는 에이전트보다 다양한 환경에 적절히 대처할 수 있는 지능적인 에이전트를 생성하는 데 중점을 둔다. 이러한 지능형 에이전트의 기본적 행동들 중에서 특히 이동(movement)을 중심으로 행동의 다양성과 효율성을 구체화해보고자 한다. 이동을 위하여 인간이나 동물 등의 생물학적 객체는 일련의 근육을 함께 사용하여 걷거나, 기거나, 뛰는 동작을 중심으로 기본적 동작들을 구현하고자 한다[6]. 실제 가상 인간(cyber-human)의 움직임은 가상s세계속의 지표면(base)의 특성과 환경 요소(Environment Factor)들에 따라 다양한 반응 양상을 보이기 때문에 지표면에 대한 모델링과 함께 환경 요소들과 인간의 관련 특성들을 종합적으로 고려해야 한다.

가상 인간의 움직임을 실세계와 유사하게 표현하기 위해서 우선 객체클래스들과 그들의 속성, 그리고 행위 등을 구조적으로 표현해주는 온톨로지를 구축한다. 그리고 지형을 중심으로 한 주변 환경의 모델링과 함께 가상 세계를 구성하는 객체들을 구체적으로 모델링하고 사람의 특성을 정의하여 이를 동작의 파라미터들로 사용함으로써 고유한 동작을 생성하고, 주변의 변화에 능동적으로 적응하여 동작할 수 있는 가상 에이전트를 구축한다. 이를 바탕으로, 지형의 기울기와 물의 깊이에 따른 행동 변화, Human의 속성과 지표면의 특성에 따른 다양한 움직임 변화, 사다리를 이용한 이동 및 장애물을 넘기 위한 점프 그리고 물체가 땅이 닿을 때 지표면과 물체의 특성을 고려한 다양한 충돌 반응 등의 Human의 행동을 구성하는 핵심요소 기능들을 구체적으로 설계하고 이들의 기능들을 종합적으로 적용하여 walk(), eat() 등의 기본 행위들을 구현함으로써 가상환경에서의 가상인간의 다양한 행동의 전체적인 개발 체계를 완성한다. 한편, 기본 행위는 기초 동작(primitive motion)들의 조합으로 만들어 진다. 이러한 방법은 기초 동작들의 재사용성에 바탕을 두고 있으며, 이는 실제로 인간의 다양한 행위를 실행하는 방법과 유사하다. 이렇게 함으로써 다양한 행동의 생성과 구현을 효과적으로 할 수 있는 장점이 있다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구들에 대해 알아보고, 3장에서는 행동(action)의 설계를 위해 행동 계층구조(Action Hierarchy)를 구성하고, 4장에서는 행위에 관련하여 인간(Human)을 모델링하고 5장에서는 인간의 행동무대로서의 가상환경을 모델링하고, 6장에서는 본 논문에서 제시한 이론들을 바탕으로 인간의 행위들을 구현하고 마지막 7장에서는 결론과 향후 연구 방향을 언급할 것이다.

II. 관련 연구

2.1 Humanoid

Humanoid 로봇은 인간들과 밀접하게 상호작용하기 위해 디자인 및 개발 되고 있으며, 인간과 유사하게 행

동하는 능력이 중요한 연구과제 중 하나이다[2]. 최근에 조명을 받고 있는 지능형 휴머노이드 로봇의 연구는 이족 보행, 시각 인식, 음성 인식 등 인간과 교감하고 지능적인 움직임을 주제로 개발되고 있으며, 이러한 연구의 일환으로 가사보조, 의료, 완구, 예술, 노인도우미 등 다각적인 응용을 시도 중에 있다[7]. 이러한 기능을 만족스럽게 수행하기 위해서는 예측할 수 없는 환경에 반응할 수 있는 능력을 갖추어야 하며, 현재와 같이 단순 반복 작업이나 일정한 프로그램에 따라 움직이는 기능만으로는 작동의 한계를 지닌다[8].

2.2 컴퓨터 애니메이션

컴퓨터 애니메이션의 궁극적 목표는 실제 사람을 닮은 가상 인물을 제작하는 것이며, 지금까지 시각적 효과에 혁신적인 변화를 일으켜왔다[4]. 기본적인 동작을 시각적으로 생성하는 일반적인 기법에는 아이콘을 이용하는 방식이 있다. 이 방식은 인체 동작에 영향을 미치는 요소들을 조절할 수 있는 아이콘을 제공하고 이들의 조합을 통해서 동작을 생성하며, 아이콘 사이에 상호작용을 통한 동작의 생성이 가능하다. 이 밖에 인간의 움직임을 만들어내는 가장 자연스러운 방법으로 모션 캡처(Motion Capture)가 있다. 움직임을 추적하기 위한 센서를 사용하여 인간의 움직임을 직접 캡처하여 삼차원의 정보로 표현하는 방식이다. 이러한 연구들은 움직임을 표현함에 있어서 역학적인 안정성과 실물과의 유사성에 그 기반을 둔다. 물리적으로 정확하고 시각적으로 만족스러운 동작의 미묘한 차이를 묘사하기 위하여, 힘과 토크의 계산, 전방 운동학(Kinematics) 사용 등 다양한 기법들이 사용된다[5]. 하지만 이러한 연구 기법들은 단지 몇 가지의 동작들의 생성에 제한되어져 있어 다양한 동작 형태를 생성하는데 한계가 있다. 그러나 본 논문에서는 에이전트의 움직임을 표현하기 위해 개체들의 속성과 행동 등을 나타내주는 지식베이스인 온톨로지를 구축하고, 기초 동작들을 정의하고 이를 조합하여 재사용함으로써 다양한 행동들을 생성할 수 있다.

III. 행동(Action)의 설계

실제인간들과 마찬가지로 가상환경 속에서 가상 인간은 시간에 따라 상황 속에서 변화를 가져오는 여러 가지 행동(action)들을 하고, 각 행동들은 가상 인간들의 기초적 동작(primitive motions)들을 구성요소로 한다. 다시 말해 가상 인간들의 기초적 동작들을 조합하여 복합적 행동이나 나아가 행사(event)들을 수행하게 되는 것이다.

3.1 Class Hierarchy

가상 환경 속에서 에이전트의 움직임을 표현하기 위해서 개체들의 속성과 행동 등을 나타내주는 지식베이스인 온톨로지를 구축하고, 이를 상황에 맞게 구체화함으로써 에이전트의 다양한 움직임을 표현할 수 있다. [그림 1]에서 보는 바와 같이 본 논문에서 제시되는 온톨로지의 최상위 레벨은 Physical Entity로 하고, Physical Entity는 Physical Object, Physical Concept, Matter로 분류한다[9].

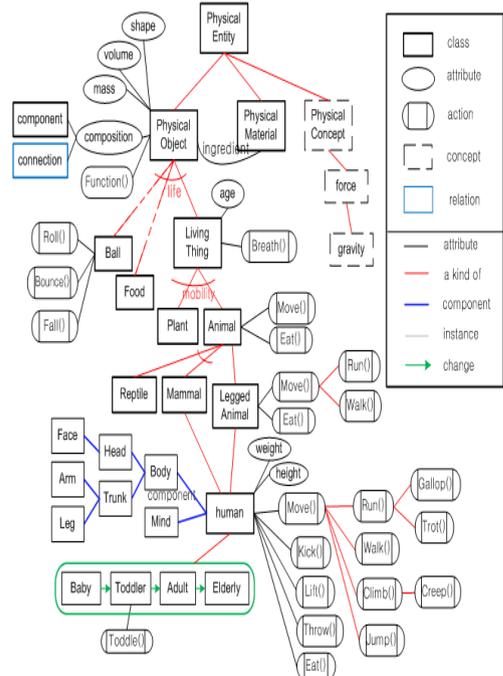


그림 1. Human의 move와 관련된 Class Hierarchy

Human Class는 크게 Body와 Mind로 구성되어 있으며, Body는 다시 Human의 각 부분들로 나누어진다. Human Class는 Age의 값을 기준으로 Baby, Toddler, Adult, Elderly로 분류되어 각각 고유한 특성을 갖는다. 각 클래스들의 속성(properties) 중의 하나인 행동들도 그들끼리 개체나 관계 클래스와 같이 계층적으로 분류할 수 있다. 이러한 'Action 계층 구조'가 다른 개체나 관계와 행위의 파라미터로서 연결됨으로써 행위의 다양화를 연출할 수 있다. 본 논문에서는 Human의 행동들을 Run(), Walk(), Climb(), Jump() 으로 세분화 가능한 Move()와 그리고, Kick(), Lift(), Throw(), Eat() 등을 중심으로 기술된다. Human Class는 Physical Entity로부터 시작해서 Legged Animal 까지 상위 클래스들을 가지며 이들로부터 상속을 받을 수 있다.

3.2 행동 계층구조(Action Hierarchy) 및 상속

행동들은 그것들을 수행하는 주체(host)와 개체에 따라서 계층구조(hierarchy)로 조직될 수 있다. [그림 2]는 Move()를 중심으로 한 행동계층구조를 보여주고 있다.

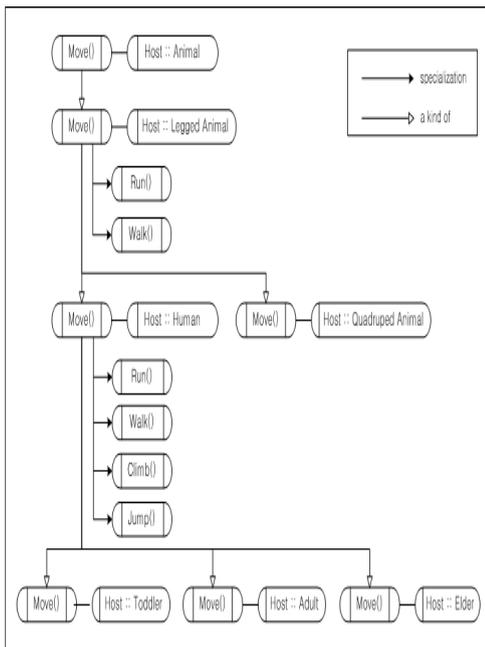


그림 2. Move()를 중심으로 한 행동계층

호스트에 따른 계층구조는 클래스 계층구조와 관련하여 구성되어 있다. 즉, 행동들의 호스트들 간의 관계가 'is a kind of'이면, 행동들 간에 관계도 'is a kind of'가 된다. 결국, 호스트가 Human인 'move()'는 호스트가 Legged Animal인 'move()'로부터 상속을 받을 수 있게 된다.

한편, 각 행동은 동일한 호스트를 가지는 세부행동들(sub-actions)로 구체화될 수 있다. 이러한 구체적 분류는 행동에 관련된 호스트의 특성 혹은 환경 요소의 차이에 의해 발생된다. 세부행동들은 그들의 특성에 따라 고유한 이름을 가지기도 한다. 즉, 호스트가 Human인 'move()'는 환경 요소 혹은 실행하는 모습 등의 차이에 따라 run(), walk(), climb(), jump() 들로 분류된다. 예를 들어, 언덕을 오르고 있다면 climb(), 구덩이가 있다면 jump()를 각각 선택하게 될 것이다. 그리고 run()은 추가적으로 다리가 지면에 닿고 있는 시간과 보폭 등에 따라 trot()과 gallop()으로 세분될 수 있다.

위에 언급된 행동들은 호스트와 지면의 상태에 의해 영향을 받게 된다. 예를 들어, Walk()을 수행하는 Human의 경우, Adult와 Elder의 걷는 모습이 다를 것이며, 경사면과 평지에서의 걷는 모습 또한 다를 것이다. 하지만 Kick(), Lift(), Throw()와 같은 행동들은 영향을 받는 대상(role)도 추가적으로 고려해야 한다. 그러한 대상들 중 본래의 역할이 특정 행동에 고정되어 있을 경우 이런 개체를 오브젝티브 호스트(Objective Host)라 부른다. 예를 들어, Ball은 Kick()의 경우 오브젝티브 호스트가 될 것이고, Ball의 무게와 위치, 크기에 따라 Kick()은 다양한 모습이 연출될 수 있을 것이다. 에이전트가 Ball을 Kick()하기 위해서는 오브젝티브 호스트의 특성과 상태를 추가적으로 고려해야 한다.

일반적인 클래스 계층구조(Class Hierarchy)에서 클래스 간에 상속은 계층 구조 내의 하위 클래스가 상위 클래스의 모든 속성(property)들을 단순히 물려받음으로 이루어질 수 있다. 이와 달리 '행동 계층구조'에서의 각 행동들은 행동 프레임(behavior frame)을 상위 클래스의 행동들로부터 상속받는다. 각 행동 프레임은 전제조건(Preconditions), 절차(Procedures), 효과(Effects) 부분들로 이루어져 있다. 전제조건은 행동 수행을 위한 필요한 조

건을 의미하고, 절차는 행동의 진행 과정을 지칭하며 일반적으로 다수의 세부행동들로 구성된다. 효과는 절차에 의해 발생하는 일련의 변화를 포함하는 결과들을 가리킨다. Walk()는 다리를 들고(Raise()), 뻗고(Spread()), 내딛는(Step()) 동작으로 구성되며, 각기 해당하는 도메인을 가진다.

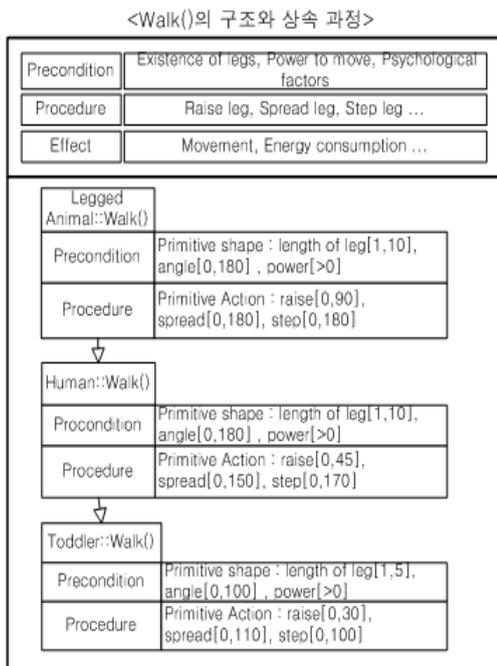


그림 3. Walk()의 구조와 상속 과정

[그림 3]은 Human의 walk()의 구조와 상속 받는 과정을 나타내고 있다. 여기서 ‘[]’는 도메인의 범위를 의미한다. 상속과정은 하위 레벨로 내려가면서 각 클래스에 맞게 행동 프레임의 각 항목에 특이성이 추가되고 도메인 범위가 축소된다.

3.3 기초 동작(Primitive motions)

본 논문에서는 walk(), climb(), jump(), eat() 등의 행동들을 사람들의 기본행위(basic actions)이라 부르고, 이것은 기초 동작(primitive motion)들의 조합으로 만들어진다. 각 기본행위들을 구성하는 기초 동작들이 [표 1]에 주어져 있다.

표 1. 기초 동작들의 예

| walk() | | climb() | | jump() | |
|-----------|----------|-----------|----------|---------|--------|
| Arm | Leg | Arm | Leg | Arm | Leg |
| raise() | raise() | raise() | raise() | raise() | sit() |
| stretch() | spread() | stretch() | spread() | pull() | rise() |
| pull() | step() | pull() | step() | | |

| eat() | | throw() | | kick() | |
|-----------|-----------|-----------|--|-----------|--|
| Arm | Mouse | Arm | | Leg | |
| stretch() | bite() | pull() | | pull() | |
| hold() | chew() | stretch() | | stretch() | |
| pull() | swallow() | | | | |

기초 동작을 조합하는 방법으로 기본행위들을 만드는 이유는 기초 동작들의 재사용성에 있으며, 이는 실제로 인간의 다양한 행위를 실행하는 행태와 유사하다. 위의 그림을 보면 같은 기초 동작이 여러 행동들에 공통적으로 나타나는 것을 볼 수 있다. 예를 들어, walk()와 climb()은 raise()라는 같은 기초 동작들을 공통적으로 사용하고, walk()의 raise()는 팔(Arm)과 다리(Leg)에 의해서 함께 사용되어 진다. 이렇게 함으로써 다양한 행동의 생성과 구현을 용이하게 하는 장점이 있다. 이처럼 공동으로 사용되어지는 기초 동작들은 파라미터의 값의 변화에 따라 여러 형태의 동작 표현에 사용될 수 있다. 그리고 각각의 기초 동작들이 정해진 순서에 따라 수행되면 전체 행동이 실행되게 된다. 예를 들어, walk()의 경우 raise(), spread(), step()이 각각 순서대로 실행되고 eat()의 경우에는 stretch(), hold(), pull()이 각각 순서대로 실행된다.

IV. Human Class

4.1 Human Class 의 모델링

인간클래스(Human Class)는 기본적으로 구성(Composition), 서술적 속성(Descriptive attributes), 기능(Function)의 항목들로 규정되어 지고, 각 항목들은 상위 클래스들로부터 상속받게 되고, 하위 클래스로 내려가면서 각각의 특이성이 추가될 수 있다. 구성은 부품들과 그들 사이의 연결로 표현되고 형태(Configuration)도

추가로 기술된다. Human에서 부품은 신체의 각 부분들이고, 연결은 각 부분들을 연결해 주는 역할을 한다. 서술적 속성은 변역(domain)이 정해져 있는 값들로 구체화되고, 기능은 각 부품들의 기초 동작들로 이루어져 있다.

```

Human = Composition(component<body, mind>, connection,
configuration),
Descriptive Attribute {human.age, human.sex,
human.power}, Functions
Body = Composition(component<head,trunk,(limb:arms
,legs)>, conection(between:head&trunk,by:neck;
between:trunk&arms,by:shoulder;
between:trunk&legs,by:pelvis;))
Descriptive Attribute {body.height, body.weight}
Functions {creep(), toddle(), walk(), stride(), run(), trot(),
hop(), jump()}
Trunk = Composition
Descriptive Attribute {trunk.length, trunk.weight}
Function {stoop(), lean()}
Arms = Composition(component<upperarm,forearm,hand>,
connection(between:upperarm&forearm,by:elbow;
between:forearm&hand,by:wrist;))
Descriptive Attribute {arm.length, arm.shape}
Functions {raise(), pull(), stretch()}
Hand = Composition(component<thumb,finger,palm>
connection(between:finger&palm,by:joint;
between:thumb&plam,by:joint;))
Descriptive Attribute {hand.length, hand.shape}
Functions {release(), hold(), grasp()}
Legs = Composition(component<thigh, fibula, foot>,
connection(between:thigh&fibula,by:knee;
between:fibula&foot,by:ankle;))
Descriptive Attribute {leg.length, leg.shape}
Functions {raise(), spread(), step()}
Foot = Composition(component<toe, instep>
connection(between:toe&instep,by:joint;))
Descriptive Attribute {foot.length, foot.shape, foot.size}
Functions {tread(), prop()}.
where ':', '[', '{', and '< >' are specialization, domain,
set and a set of components.
    
```

그림 4. Human 클래스의 모델링

인간클래스는 [그림 4]와 같이 정의할 수 있으며, 인간의 움직임은 인간클래스의 부품으로 규정된 각 부분들의 기능의 조합으로 이루어진다. 기능들은 인간의 움직임을 표현하기 위한 기초 동작들과 기본행위들이 된다. 본 논문은 객체(Object)의 이동과 함께 인간(Human)의 행위를 표현하는 것에 그 중점을 둔다. 이를 위해 인간의 각 요소들은 질량과 크기 등의 속성을 가지는 객체의 하위 클래스들이므로, 그것들의 움직임은 객체의 행동에 의한 상속과 인간 클래스가 가지는 특이성을 추가하여 표현하게 된다. 본 논문에서 가상 인간들은 지면의 상태와 인간의 나이, 건강상태 등의 다양한 요소들을 고려하여 Human의 적절한 행동을 선택할 것이다.

4.2 Human body의 구조

인간의 신체(Human Body)는 각 부위들과 그것들을 연결하는 관절들로 구성 되며, 관절들은 목, 허리, 어깨, 팔꿈치, 손목, 골반, 무릎, 발목 그리고 손과 발에 대한 관절들로 구성되어 있다. 그리고 각각의 연결부위(joint)들은 특성에 따라 1에서 3까지의 다양한 자유도(degree of freedom)을 가진다. [그림 5]는 신체를 구성하는 여러 관절들에 대한 개념도이다.

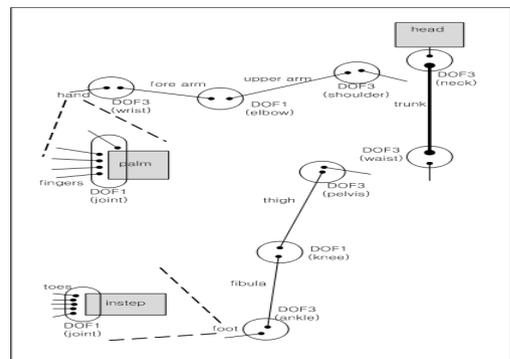


그림 5. 인간 신체의 구조

4.3 팔 다리의 움직임 모델링

앞에서 언급된 구조를 바탕으로 팔과 다리의 움직임을 x, y 축을 가진 2차원 그래프로 표현해 보면 다음과 같다[10]. 여기서 팔과 다리의 길이는 상부링크(l_1)와

하부링크(l_2) 두 부분으로 나누고 각각이 이루는 각도들(θ_1, θ_2)과 길이에 따른 x, y 값을 아래와 같이 식(1)로 계산할 수 있다.

$$\begin{aligned} x &= l_1 \cos \theta_1 + l_2 \cos (\theta_1 + \theta_2) \\ y &= l_1 \sin \theta_1 + l_2 \sin (\theta_1 + \theta_2) \end{aligned} \quad (1)$$

팔의 경우엔 x 는 뻗을 수 있는 길이, y 는 팔의 직선 길이가 되며, 다리의 경우엔 x 는 뻗었을 때의 보폭, y 는 다리의 직선 길이가 된다.

이 수식을 이용하여 속도의 개념을 정의할 수 있는데 위의 식을 시간 t 로 미분을 하면 다음과 같이 식(2)가 된다.

$$\begin{aligned} \frac{dx}{dt} &= \frac{d\{l_1 \cos \theta_1 + l_2 \cos (\theta_1 + \theta_2)\}}{dt} \\ \frac{dy}{dt} &= \frac{d\{l_1 \sin \theta_1 + l_2 \sin (\theta_1 + \theta_2)\}}{dt} \end{aligned} \quad (2)$$

이를 정리하면 아래와 같이 식(3)으로 나타낼 수 있다.

$$\begin{aligned} \dot{x} &= \{-l_1 \sin \theta_1 - l_2 \sin (\theta_1 + \theta_2)\} \dot{\theta}_1 + \{-l_2 \sin (\theta_1 + \theta_2)\} \dot{\theta}_2 \\ \dot{y} &= \{l_1 \cos \theta_1 + l_2 \cos (\theta_1 + \theta_2)\} \dot{\theta}_1 + \{l_2 \cos (\theta_1 + \theta_2)\} \dot{\theta}_2 \end{aligned} \quad (3)$$

위의 식(2)(3)을 행렬식으로 표현하면 식(4)와 같다.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} -l_1 \sin \theta_1 - l_2 \sin (\theta_1 + \theta_2) & -l_2 \sin (\theta_1 + \theta_2) \\ l_1 \cos \theta_1 + l_2 \cos (\theta_1 + \theta_2) & l_2 \cos (\theta_1 + \theta_2) \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \quad (4)$$

이 식(4)이 [그림 6]의 2자유도 평면운동의 순간운동 공식이 된다.

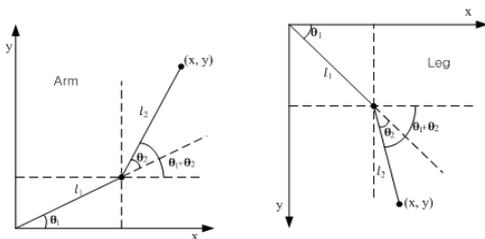


그림 6. Human 팔과 다리의 2자유도 평면운동

관절속도에 자코비안 행렬 J 을 곱함으로써 (x, y) 좌표에 대한 속도를 구할 수 있다. 본 논문에서는 위의 공식을 바탕으로 걷기에 관련된 기초 동작들을 만들어 내고 이를 조합하여 다양한 Human 행동들을 표현한다.

V. 가상 환경 모델링

Smile()등과 같은 내부적 행동들과 달리 Human의 Walk(), Run(), Lift(), Throw(), Eat() 등의 외부영역이나 외부객체와 관련된 외부적 행동들의 표현을 위해서는 행위주체 외에 추가로 고려해야 할 요소들이 존재한다. 본 논문에서는 먼저 지면 상태와 마찰력을 중심으로 지면(Base)을 구축하고, 중력과 그 외의 힘 그리고 지면의 상태에 따라서 이동하는 객체(Object)를 모델링하여, 지면과 객체 그리고 객체의 하위 레벨 클래스인 인간(Human)들 간에 생길 수 있는 다양한 접촉 관계를 표현하는데 초점을 맞춘 가상 환경을 구축하고자 한다. 중력과 마찰력 등 자연현상을 일으키는 힘은 눈에 보이지 않기 때문에 이를 표현하기 위해서는 이러한 힘들에 의한 물리적 객체들의 다양한 움직임이나 변화를 통해서 표현할 수밖에 없다.

5.1 지면(Base)의 모델링

지면(Base)에 대해 고려해야 할 주요 요인은 ‘마찰력’과 ‘기울기’이다. 마찰력은 지면의 재질과 상태에 의해 결정되는 마찰 계수(μ)와 중력(G), 그리고 물체의 질량(m)에 비례하여 발생된다. 본 논문에서는 지면의 재질을 크게 세 부분으로 분류한다. 지면각각이 지닌 요인들 즉 강도(solidity), 탄성(elasticity), 접착성(adhesivity) 등의 값에 따라 가장 영향력이 높은 요인을 지배 요소(dominant factor)로 잡고, 강도, 접착성, 탄성이 높은 장소의 예로 바위산, 진흙탕, 풀밭을 들 수 있다.

[그림 7]에서 보는 바와 같이 지면의 종류와 인가된 압력의 세기 따라 가장 우세를 점하게 되는 힘을 결정할 수 있다. 예를 들면, (c) meadowy base에 a 와 b 사이에 압력이 가해진다면 ‘elastic force’가 지배요인이 되어서 객체는 bounce()를 하게 된다. 또한 위의 세 가

지 힘은 충돌하는 객체의 성질에 따라서도 달라진다. 예를 들어 지면의 'elastic force'는 객체의 속도와 탄성에 비례하는 성질을 가지게 되는 것이다[11].

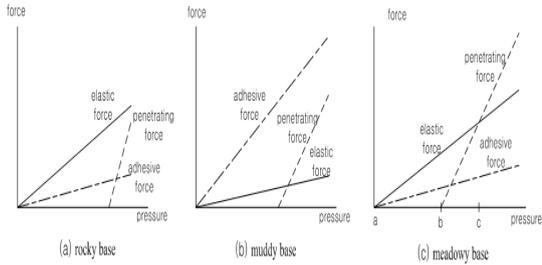


그림 7. Base의 압력과 힘 그래프

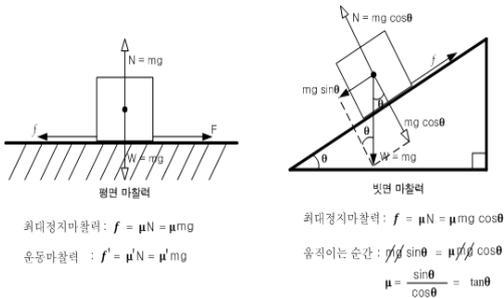


그림 8. 평면과 빗면의 마찰력

[그림 8]에서 보는 바와 같이 물체가 움직이는 바닥은 평면이거나 기울기를 가진다. 평면에서 최대 정지 마찰력은 중력 또는 수직항력의 크기에 비례하고 지면의 재질을 나타내는 상수인 정지마찰계수(μ)에 비례한다. 운동 마찰력도 수직항력의 크기와 움직이고 있는 상황에서의 마찰계수인 운동마찰계수(μ')에 비례한다.

5.2 Object 모델링

```
Overall positioning = f(O.mass, O.color, O.shape, velocity(O))
mass > 0
[color] = {red | yellow | blue ....}
[shape] = {sphere | box | pyramid ....}
velocity = f(gravity, force from human, wind force)
```

그림 9. 객체를 구성하고 있는 속성

객체의 움직임을 공식화하기 위해서 [그림 9]와 같이 우선 객체를 구성하고 있는 속성들의 규정이 필요하다.

```
{shape : break | deform | sustain }
{movement() : 수직상하운동() | 포물선 운동() | bounce().....}
```

그림 10. 지면과 충돌 후 보이는 반응 양상

객체가 인간에 의해 힘이 가해지거나 지면과 충돌 후에 보이는 반응 양상은 [그림 10]과 같이 다양한 양상을 보인다.

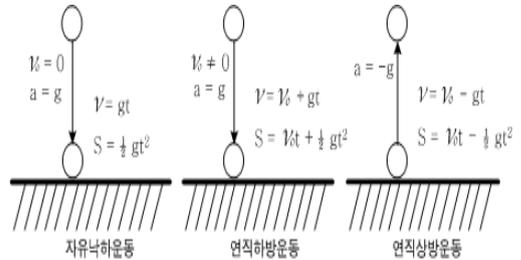
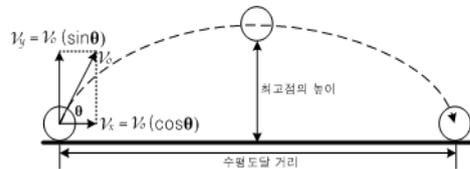


그림 11. Object의 수직 상하 운동

'shape'의 충돌에 의한 변화는 충돌 시 받게 되는 '압력'에 따라 결정되어지고, '압력'값은 힘에 면적을 나눈 값으로 계산되어 진다. 수직상하운동에는 자유낙하, 연직하방, 연직상방운동으로 나누어진다. 객체의 수직 상하 운동은 [그림 11]과 같이 나타낼 수 있다.



- t초 후의 x축 방향의 위치 S_x : 등속도 운동 $S_x = V_0(\sin\theta) t$
- t초 후의 y축 방향의 위치 S_y : 중력가속도가 작용 $S_y = V_0(\sin\theta) t - 1/2gt^2$

그림 12. Object의 포물선 운동

그리고 Throw()에 대한 객체의 움직임을 [그림 12]

와 같은 포물선 운동으로 나타낼 수 있다.

VI. 구현 및 실험결과

6.1 구현의 전체적인 흐름

[그림 13]은 구현 시스템의 전체적인 흐름도를 나타낸 것이다.

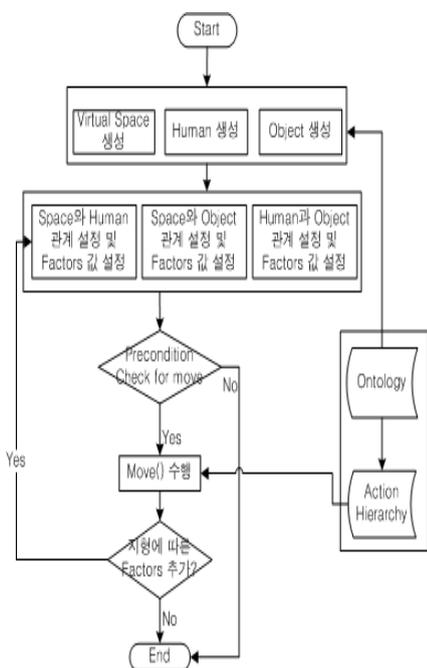


그림 13.move()의 구현을 위한 전체적 순서도

가상세계 속의 공간에 거주하는 에이전트는 데이터베이스에 저장된 가상세계에 관한 자신의 지식을 활용하여 판단을 하게 된다. 이러한 지식을 이용하여 인간(인스턴스)가 여러 가지 지표면과 다양한 장애물이 포함된 환경에 적응하여 움직이는 모습을 확인할 수 있다.

6.2 Walk() 의 매개변수들

행동들 중 Walk()의 매개변수들과 그들과의 상호 관계를 살펴본다. [그림 14]는 Walk()의 매개변수들 간의 상호 관련들을 나타내고 있다.

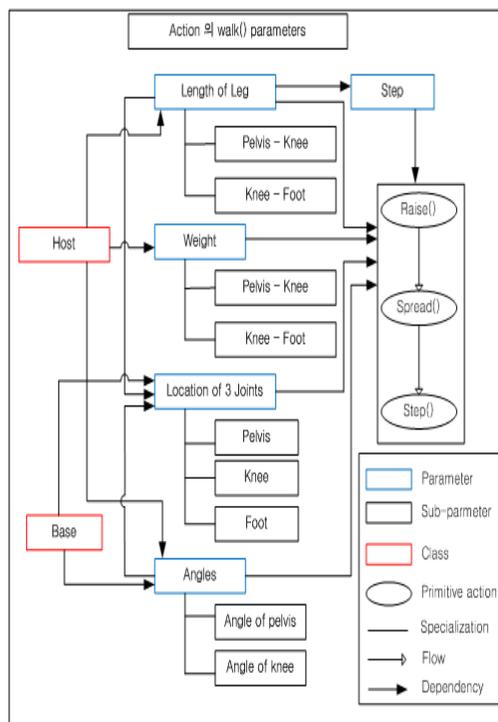


그림 14. walk()의 매개변수사이 관계도

Walk()를 수행하기 위해 다리의 길이(Length), 무게(Weight), 위치(Location), 각도(Angle) 들이 고려되어야 하며 각 변수는 행동들을 수행하는 호스트(Host)에 따라 그 값이 달라지며, 행동과 지면의 종류에 영향을 받는다. 그리고 다리의 길이는 Human 의 보폭(Step)의 값을 결정한다.

6.3 구현 및 결과

앞에서 제시한 이론들을 통해서 인간(Human)의 행위들을 구현한다. 구현 환경은 MS의 Windows에서 Visual C++ 6.0과 OpenGL API를 응용하여 제작하였다. [그림 15]은 화면에서 Human의 동작을 나타내고 지면의 상태와 오브젝트를 나타내는 초기화면으로 3차원 그래픽으로 구현하여 인간이 취하는 동작과 자세를 여러 가지 다른 시각들에서 바라볼 수 있게 구현하였다.

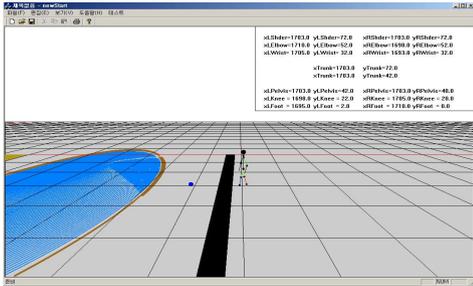


그림 15. 시뮬레이션 화면

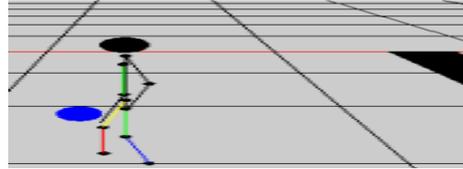


그림 17. 공을 들고 걷는 모습

아래 [그림 18]에서는 지면이 진흙으로 이루어진 부분에서 Human이 공을 떨어뜨리는 화면이다..

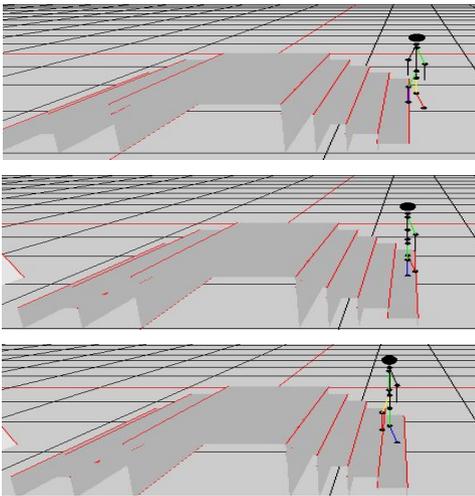


그림 16. 계단 올라가는 모습

[그림 16]은 지면에 계단을 생성하여 Human이 계단의 폭과 높이에 따라 보폭과 각도를 조절하여 계단을 올라가는 모습을 확인할 수 있다.

다음은 지면의 상태에 따라 Human의 걷는 모습의 다양화를 구현해 보았다. [그림17]은 Human이 물건을 들고 걷는 모습이다.

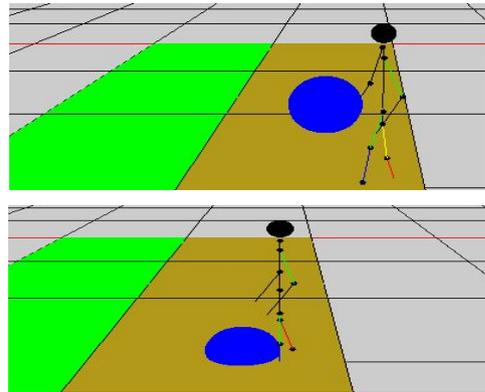


그림 18. 진흙에 공이 묻는 화면

아래 [그림 19]에서는 탄성이 있는 풀밭 부분을 나타내는 것이다. Human이 들고 있는 공을 떨어뜨렸을 때 지면의 성질에 따라 지면에 묻히거나 튀는 현상을 보여 주고 있다.

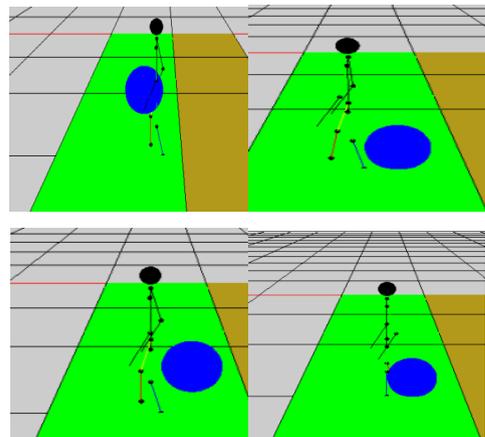
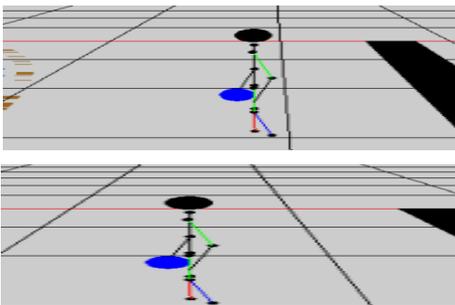


그림 19. 풀밭에 떨어지는 공이 튀는 화면

다음은 경사면의 상태에 따라 발생하는 행동 변화를 표현해 보았다. [그림 20]은 Human이 진흙으로 이루어져 있는 경사면을 걷는 모습이다.

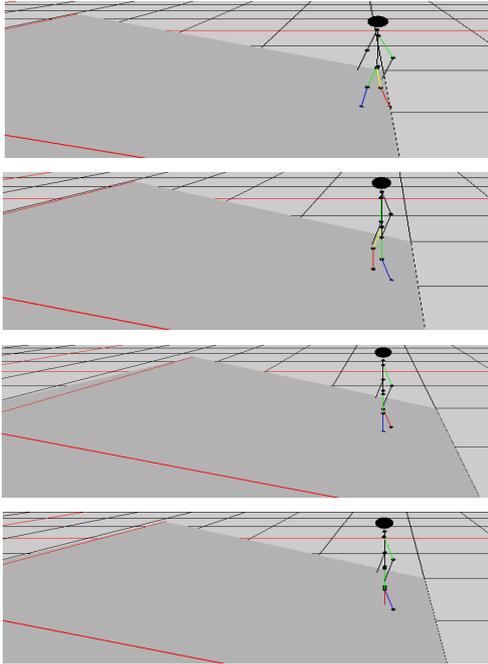


그림 20. 진흙으로 이루어진 경사면을 걷는 모습

Human이 진흙에서는 몸무게 때문에 걸으면서 발이 진흙에 약간 묻히는 모습을 확인해 볼 수 있다.

아래 [그림 21]에서는 화면에 담을 생성하고 사다리를 이용하여 올라가는 행동을 구현한 내용을 보여준다.

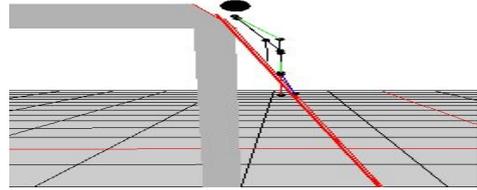
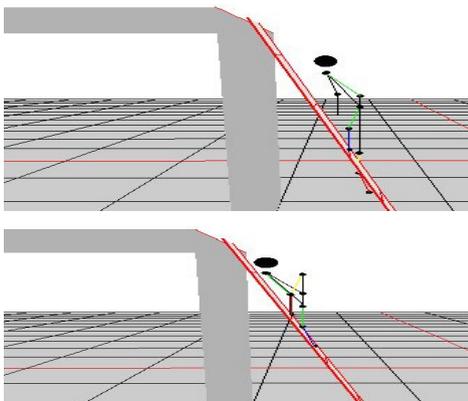


그림 21. 사다리 올라가는 모습

사다리가 놓여 있는 경우 사다리를 손으로 잡고 사다리의 간격에 따라 인간의 보폭을 결정하여 사다리 올라가는 행동을 취하게 된다.

Ⅶ. 결론 및 향후 연구방향

기존의 연구에서는 가상세계에 하나의 에이전트에 국한된 객체의 생성과 움직임을 표현에 중점을 두었다. 가상 세계의 다양한 객체들의 움직임을 표현하기 위해서는 에이전트들 간의 관계들을 포괄할 수 있는 모델링 기법이 필요하며, 또한 복수의 가상 에이전트들 간에 발생할 수 있는 다양한 효과들을 규명하는 작업이 필요할 것이다.

본 논문에서는 가상세계에서 에이전트와 객체들 그리고 지면에 대해 모델링하고 이에 따른 다양한 움직임을 위한 모델링과 표현 방법에 대해 알아보았다.

이를 위해 객체들의 속성과 행동 등을 나타내주는 지식베이스인 온톨로지를 구성하였고, 특히 물리적 객체를 비롯한 인간의 속성들을 구체적으로 정의하였다. 최상위 객체들로부터 개별 에이전트까지 계층구조에 기반을 두어 상속을 통하여 수많은 행위들을 쉽게 구현할 수 있다. 그리고 소수의 기초 동작들을 정의하고 이들을 재사용함으로써 에이전트의 다양한 행위들이나 객체들의 동작들을 용이하게 구현할 수 있다.

이를 통해 가상 환경에서 일어날 수 있는 인간의 많은 움직임들을 표현할 수 있을 것이며, 나아가 사람들이 그럴듯하다고 느낄 수 있는 가상현실을 만드는데 용이한 방법론이 될 것이다.

참 고 문 헌

- [1] S. J. Ji, "A Structured Causal Graph Based Model for Event Development," Computer and Advanced Technology in Education 2001, Banff, Canada, 2001.
- [2] K. Harada, K. Hauser, T. Bretl, and J. -C. Latombe, "Natural Motion Generation for Humanoid Robots", Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, October 9-15, 2006, pp.833-839.
- [3] Norman I. Badler and Diane M. Chi, "Virtual Human Animation Based on Movement Observation and Cognitive Behavior Models," Computer Animation, 1999. Proceedings, pp.128-137, 1999(5).
- [4] Rick Parent, "컴퓨터 애니메이션 - Algorithms and Techniques", 아진출판사, p.34, 2004.
- [5] A. Bruderlin and T. Calvert, "Goal-Directed, Dynamic Animation of Human Walking," Computer Graphics(Proceedings of SIGGRAPH 89), pp.233-242, 1989.
- [6] Alex J. Champandard 저, 이강훈 역, "인공지능 게임 프로그래밍 실전 가이드:최신 AI기법을 적용한", 에이콘출판사, p.72, 2005.
- [7] 권오상, "지능형 홈로봇 기술 전망", (주)한울로보텍스, 2003.
- [8] K. Park, "Rapid Development of a Humanoid Robot using CAD/CAM/CAE/FP", 한국 CAD/CAM 학회 학술 발표회, 2006.
- [9] J. H. Park, "Modelling Cosmic Elements for Cosmic Simulation", Tech. Report #97, AIMM Lab, Kyungpook Nat'l Univ., 2007.
- [10] Kadota, Kazuo. 김진오 저, "로봇공학의 기초", 성안당, 2008.
- [11] B. J. Kim, "A Logical Model of Collision Response for Simulation of the Virtual

Environment," 경북대학교 석사 학위논문, 2003.

저 자 소 개

김 유 신(Yu-Shin Kim)

정회원



- 2006년 8월 : 안동대학교 컴퓨터 교육과(이학사)
- 2011년 2월 : 경북대학교 모바일 통신공학과(공학석사)
- 2011년 2월 ~ 현재 : (주)무비소프트 근무

<관심분야> : 인공지능, 시뮬레이션 분야

정 근 재(Geun-Jae Jung)

정회원



- 2005년 2월 : 대구대학교 전산통계학전공(이학사)
- 2007년 8월 : 경북대학교 정보통신학과(공학석사)
- 2007년 8월 ~ 현재 : 경북대학교 전자전기컴퓨터학부 박사과정

<관심분야> : 인공지능, 인간과 컴퓨터 상호작용, 가상현실

박 종 희 (Jong-Hee Park)

정회원



- 1979년 : 서울대학교(공학사)
- 1981년 : 한국과학원(공학석사)
- 1990년 : Univ. of Florida(공학박사)
- 현재 : 경북대학교 전자공학과 교수

<관심분야> : 멀티미디어 응용, Computer Aided Education, CAD/CAM, 지능형 정보 시스템, 분산데이터 처리 시스템