

# 가상 세계 거주자의 지각 메커니즘 설계 및 구현

## Design and Implementation of the Perception Mechanism for the Agent in the Virtual World

박재우\*, 정근재\*, 박종희\*\*

경북대학교 전자전기컴퓨터공학부\*, 경북대학교 전자공학과 정보통신학전공\*\*

Jae-Woo Park(pjw98@korea.com)\*, Geun-Jae Jung(cooky8884@ee.knu.ac.kr)\*,  
Jong-Hee Park(jhpark@ee.knu.ac.kr)\*\*

### 요약

가상 세계에서 인간과 유사한 에이전트를 만들기 위해서는 지각, 인식, 판단 그리고 행동에 대한 정교한 설계가 중요하다. 이와 관련하여 자율형 에이전트의 지각 기능과 인식 기능을 개발하고자 한다. 시야 속에서 획득되어진 가장 원시적 데이터인 이미지의 색상차를 이용하여 모양과 영역들을 구별하는 것으로부터 점, 선 색깔들을 기본 단위로 사용하는 지각 메커니즘을 개발한다. 이렇게 지각되어진 모양들을 지능적으로 인식하기 위해서 가려지거나 손실된 모양에서 원래의 모양을 추측하기 위한 추론 알고리즘을 개발하고 객체에 관해 온톨로지로 부터 얻어진 일반적 특성정보를 이용한다. 개별적으로 파악된 이차원 모양들과 다른 모양들과의 공간적 위치관계들이 삼차원 모양들을 이루고 그러한 모양을 가진 해당 객체들은 장면들을 구성하게 된다. 삼차원 모양들은 각 장면에서 자신만의 영역을 차지하며 에이전트는 객체들과 장면들을 분석하여 사물과 현상들을 인식한다. 이러한 장면에 대한 인식기능을 이용하여 에이전트가 시공간 영역 속에서 지식을 축적하고 이용하는 방법을 개발하고 예제상황을 통해 구현결과를 보여준다.

■ 중심어 : | 지각 | 인식 | 온톨로지 |

### Abstract

In order to create an intelligent autonomous agent in virtual world, we need a sophisticated design for perception, recognition, judgement and behavior. We develop the perception and recognition functions for such an autonomous agent. Our perception mechanism identifies lines based on differences in color, the primitive visible data, and exploits those lines to grasp shapes and regions in the scene. We develop an inferencing algorithm that can infer the original shape from a damaged or partially hidden shape using its characteristics from the ontology in order to intelligently recognize the perceived shape. Several individually recognized 2D shapes and their spatial relations form 3D shapes and those 3D shapes in turn constitute a scene. Each 3D shape occupies its respective region, and an agent analyzes the associated objects and relevant scenes to recognize things and phenomena. We also develop a mechanism by which an agent uses this recognition function to accumulate and use her knowledge on the scene in the historical context. We implement these functions presented above against an example situation to demonstrate their sophistication and realism.

■ keyword : | Perception | Recognition | Ontology |

\* 이 논문은 BK21 경북대학교 인력양성 사업단과 경북대학교 산학협력단의 지원을 받아 연구하였습니다.

접수번호 : #110408-005

접수일자 : 2011년 04월 08일

심사완료일 : 2011년 07월 29일

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

## 1. 서론

가상 세계에서 에이전트는 자율적인 행동을 수행하는 능력을 가진 가장 중요한 개체이다. 가상 세계의 무대를 이루는 가상공간은 단순한 그래픽이 아니라 부피, 중력, 마찰력과 같은 물리적 속성들과 법칙들이 적용되는 질서 있는 공간을 의미 한다[1]. 에이전트는 이러한 가상공간 속에서 활동하기 위해서 주변을 지각하고 인식하여 상황을 파악한다[2].

가상 세계를 응용하여 개발된 기존의 게임들은 에이전트와 유사하게 디자인 되어진 캐릭터들은 대부분 자신과 관련된 메시지만을 전달받아 미리 프로그램 된 행동들만 수행한다[3]. FSM(Finite state machine)은 주어지는 모든 시간에서 처해 있을 수 있는 유한개의 상태를 가지고 주어지는 입력에 따라 어떤 상태에서 다른 상태로 전환시키거나 출력이나 액션이 일어나게 하는 장치 또는 그런 장치를 나타낸 모델이다. FSM을 사용하는 게임은 에이전트의 행동 종류와 그 범위를 미리 정하고 에이전트의 행동 메시지에 반응하는 에이전트들을 구성하면서 전체 시스템을 구축해나간다[2]. 이 방식은 쉽게 에이전트들의 상태를 정의 할 수 있고 게임의 규칙들도 명확하게 확정할 수 있는 장점이 있다. 하지만 FSM으로 구현된 시스템은 사용자가 에이전트들의 반응들이 한정되어 있고 그들에게 다양한 형태의 메시지를 전달할 수 없기 때문에 쉽게 몰입감을 잃어버리는 단점을 가지고 있다. EA사에서 개발한 '십즈3'와 린드랩의 '세컨드 라이프'는 이러한 단점을 해결하기 위해 게임 내에 등장하는 에이전트들을 각기 다른 사용자들이 조종함으로써 게임의 몰입감을 유지해 나갈 수 있도록 가상 세계를 설계하였다. 이러한 상황 전개 방식도 에이전트들의 상황 인식을 인간에게 의존하는 한계가 있다.

먼저 에이전트는 자율적인 판단에 기초하여 행동하기 위해서 상황을 지각하고 인식하는 능력을 가져야 한다. 본 논문에서 에이전트들은 가상세계에서 전개되는 상황들을 시각적으로 지각하고 인식할 수 있는 기능을 시뮬레이션 하는 방법을 연구하고자 한다. 에이전트들은 지각이나 인식 기능들을 통해 얻어진 고유한 지식을

가지게 되며 이는 결과적으로 독특한 개성을 가진 자율형 에이전트(autonomous agents) 들이 만들어 지게 된다. 본 논문에서 에이전트는 스스로 지각한 데이터를 이용하여 상황을 인식한다. 기존에 개발되어진 자율형 에이전트들은 다른 객체들이 보내주는 정형화된 메시지(message)에 반응하여 상황을 파악하는 방법과는 차별화된다[4].

본 논문에서는 인식의 전제인 지각을 위해 눈에 보이는 객체에 대한 시각의 시뮬레이션에 초점을 두고 그 결과를 이용하여 객체의 인식 방법을 개발한다. 에이전트들은 같은 상황을 지각하더라도 각자의 지식과 정보의 차이로 인해 서로 다른 결과를 도출할 뿐 아니라 부정확한 인식으로 인해 엉뚱한 행동을 하는 등 실세계에서 처럼 다양한 상황을 연출할 수 있다[5]. 본 논문에서 에이전트는 시야 안에 들어온 이미지의 색상 차이를 이용하여 모양들을 구별하고 이 데이터들과 자신이 가진 지식을 비교하면서 객체와 지역을 인식한다.

기존 연구에서는 인간의 지각 방법을 바탕으로 모양의 추상화 및 정형화 작업들에 관련하여 그래픽 정보를 의미론적 데이터(semantic data)로 표현하기 위한 연구들이 진행되어 왔었다[4][6]. 그러나 기존의 연구들은 이차원 그래픽을 처리하고 그것들을 단순한 기호들의 조합으로 표현하는 방법을 사용했기 때문에 삼차원 그래픽처럼 원근감과 불투명이 있는 영상들을 처리하기에 어려움이 있다. 본 논문에서는 이차원으로 인식된 모양들을 바탕으로 삼차원 영상들을 분석하고 저장할 수 있는 방법을 제시한다. 그리고 에이전트는 이 방법들을 이용하여 가상공간에서 상황들을 지각하고 인식하는 방법들을 다룬다.

## 2. 관련 연구

### 2.1 그래픽스 인식(Graphics recognition)

그래픽스 인식은 이미지를 해석하기 위해 이미지로부터 그래픽 성분들을 추출하는 단계와 이미지에서 추출한 것들을 인식하는 단계로 나눌 수 있다[7]. 그래픽스 이미지에서 성분들을 추출하는 방법은 픽셀(pixel), 벡터(vector), 호(arc), 곡선(curve), 영역(region) 그리고 기호(symbol)와 같은 요소들을 사용한다[8-10]. 이

런 성분 추출 방법들은 이미지 데이터의 색상 차이를 통해 점, 선, 모양, 지면 등의 그래픽 성분들을 추출하여 대상이 되는 물체를 인식하는 것이 가능하기 때문에 많은 패턴 인식 분야에서 사용되고 있다. 본 논문에서도 그래픽 성분을 추출하기 위해 이러한 추출 방법들을 사용하며 이차원 모양(shape)들을 삼차원 모양으로 인식하기 위해 그래픽 성분들을 연속된 장면으로 저장한다.

### 2.2 온톨로지(Ontology)

온톨로지는 인공지능의 지식 표현(knowledge representation) 기술과 관련된 중요한 분야로서 특정 분야를 대상으로 객체의 종류 및 속성, 객체들 사이의 관계에 대해 정의한 내용(content) 이론이다[11][12]. 온톨로지는 가상 세계를 현실 세계처럼 규칙들(rules), 법칙들(principles), 제약들(constraints)의 지배를 받는 가상 환경의 제어와 에이전트들의 지식 구조와 객체들의 정의에 사용된다. 가상 세계는 많은 객체들로 이루어져 있고 존재하는 각각의 객체들은 자신만의 특성들을 가지고 있으며 동시에 다른 객체들과 다양한 관계를 갖고 있다. 온톨로지는 개체들이 계층 구조(Class Hierarchy)로 구성되어지며 각 개체(Class)들의 특성들을 상속 받아서 객체들을 규정할 수 있는 정보를 제공한다[13].

### 2.3 지능형 이동 로봇의 물체 인식

지능형 로봇(Intelligent Robots)은 외부환경을 인식하고 스스로 상황을 판단하여 자율적으로 동작하는 로봇을 의미한다. 최근 지능형 이동 로봇과 관련된 자동 지도 생성 및 위치 추정 연구(SLAM)가 활발히 진행됨에 따라 이동 로봇이 주변 환경과 상호작용을 하기 위한 물체 인식 기술이 필수 요소로 부각되고 있다[14]. 기존의 단순 로봇과 달리 지능형 이동 로봇을 위한 물체 인식 기술은 제한된 환경이 아닌 일반 환경에서 로봇이 이동하면서 주변 물체를 인식해야 하기 때문에 다양한 상황과 변수들이 발생한다[15]. 지능형 이동 로봇은 보는 위치와 조명(照明)등의 조건이 수시로 변하는 환경에서 물체를 인식할 때 제한된 정보만으로 인식을 한다. 따라서 주어진 물체 정보에 정확히 부합하지 않

는 환경에 대해 얼마나 잘 대처하는가가 인식 문제 해결의 핵심 요소이다[14]. 이와 유사하게 본 논문에서의 에이전트도 주변 환경 정보를 인식하고 대응 할 뿐 아니라 연속된 시간의 흐름에 따라 지각된 정보들을 기억해 나간다. 따라서 가상 세계의 에이전트는 많은 부분에서 이동형 지능 로봇과 유사한 인식 기능을 가지며 정교하게 디자인된 에이전트의 인식 알고리즘은 이동형 지능 로봇의 인식률을 높이는데 활용할 수 있다.

## 3. 지각 데이터의 표현

### 3.1 객체의 모양에 대한 지각 및 인식 과정

지각은 에이전트의 시야 안에 들어온 이미지의 색상 차이를 이용하여 경계(boundary)들을 추출하는 작업에서 시작한다. 이 경계들은 선 단위로 저장되며 이 선들은 이차원 모양(shape)들의 구성 성분이 된다. 이차원 모양들은 자신만의 속성 값들을 가지며 다른 이차원 모양들과 위치 관계를 형성한다. 이차원 모양들은 모양들 사이의 위치 관계에 의해 삼차원 모양을 구성하는 면(面)들로 재배치한다. 에이전트는 삼차원 모양들의 유형, 크기, 색상, 위치 관계 등의 속성 값을 자신이 가진 개체 정의와 비교하여 물체를 인식한다. 본 논문에서 지각 메커니즘은 세부 영역을 먼저 분석하고 상위 영역을 세부 영역의 관계들로 규정하는 유한 요소법(Finite Elements Method, 有限要素法)을 사용한다. [그림 1]은 유한 요소법을 사용하여 하나의 삼차원 객체를 분해하고 그것을 데이터로 정형화하는 과정을 개략적으로 보여주고 있다.

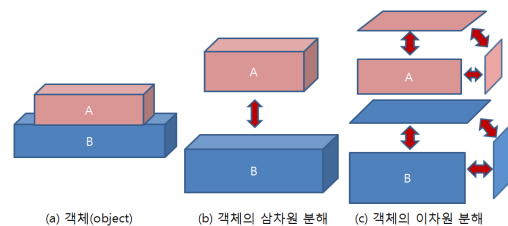


그림 1. 객체의 분해

예를 들어 에이전트가 [그림 1]의 (a)와 같은 객체를 보면 (b)와 같은 분해 과정을 거쳐 (c)에서처럼 면의 형

태로 객체의 구성 요소들을 분석한다. 에이전트는 (b)와 (c) 단계를 거치면서 데이터를 정형화하는 알고리즘을 공통적으로 적용한다. 그러나 (a)단계는 에이전트마다 다르게 적용될 수 있다. 어떤 에이전트는 (a)단계에서 객체를 모자 모양이라고 판단할 수도 있고 또 다른 에이전트는 바퀴 모양이 추상화된 자동차와 유사하다고 판단할 수 있다. 각각의 에이전트가 이런 차이를 보일 수 있는 것은 각 에이전트들이 가진 객체의 정의나 특징이 서로 다르기 때문에 발생한다.

### 3.2 장면(scene)의 지평선(Horizon)

장면은 상황의 기본단위로서 지각되어진 정보들을 장면 단위로 기록한다. 장면은 ID를 포함하여 초단위의 시간 정보와 지평선 높이에 관한 데이터들로 구성된다. 이러한 정보들을 이용하여 에이전트의 과거 기억들을 구성함으로써 장면 단위로 분석이 가능하며 과거에 일어났던 사건의 정보들을 이용할 수 있다. 또한 인접한 장면에서의 객체 위치를 분석하여 그것들의 이동 경로를 예측할 때에도 이 정보들을 사용한다.

각 장면에서 객체들의 위치를 인식할 때 그 객체가 바닥(base)의 구성 요소인지 아닌지를 판단하는 기준으로 지평선을 사용한다. 이 구분법은 삼차원 입체 영상 기법에서 사용하는 방법이다[16]. 지평선 구분법은 지평선의 아래는 바닥 영역으로 지평선의 위는 하늘 영역으로 설정하고 이 영역들 안에 나머지 객체들을 배치시키는 방법을 사용한다. 인식 단계에서는 지평선 아래에 위치한 하나의 면(面)으로 이루는 삼차원 모양을 바닥으로 정의한다. 장면에 이 방법을 이용하여 바닥들을 배치하면 그들의 위치를 더 정확하게 계산할 수 있고, 위치 계산 시간을 단축할 수 있는 장점이 있다. 지평선은 하늘과 닿은 선들 가운데 좌표  $cy$ 값(선의 중심점  $y$  값)이 가장 작은 것으로 정한다. 그러나 장애물에 의해 하늘이 보이지 않거나 에이전트가 건물 내부에 들어있는 경우 지평선이 시야 속에 존재하지 않는 경우도 발생한다. 이때는 시야 안의 중심점을 계산하고 그것의  $y$ 값을 지평선으로 사용한다. [그림 2]는 시야 안의 지평선을 정하는 예를 보여주는 그림이다.

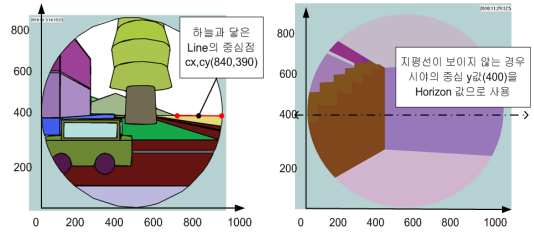


그림 2. 시야 안의 지평선 값들

[그림 2]에서 왼쪽 그림은 지평선이 존재한다. 그러나 오른쪽 그림에서는 에이전트가 건물 내부에 있다. 그렇기 때문에 지평선이 존재하지 않는다. 따라서 시야의 중심점을 이용하여 400을 지평선으로 정한다.

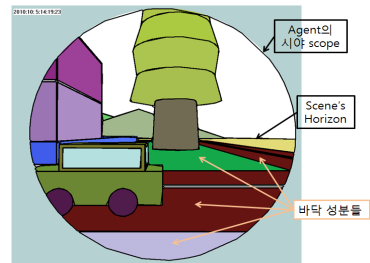


그림 3. 에이전트 시야 내부의 지평선과 바닥들

위의 [그림 3]은 에이전트가 시각안의 지평선 값을 이용하여 인식한 바닥 객체(base object)들을 보여주는 그림이다. 지평선 아래에 존재하면서 하나의 면(面)만을 갖는 삼차원 객체들을 바닥 객체들로 인식한다.

### 3.3 이차원 모양 데이터 표현

기존 연구에서 이차원 모양들의 세만틱 표현 방법과 관련된 추론 알고리즘 그리고 모양들의 위치 관계 표현 방법을 소개했다[4].

본 논문에서는 기존 연구에서 좀 더 확장하여 [표 1]과 같이 이차원 모양에 관련된 정보들을 구체적으로 정의하였다.

표 1. 이차원 모양 테이블

Field	설명
SceneNum	장면 ID
ID	이차원 모양의 ID
type	이차원 모양의 유형(Type)
color	색상 (red, green, blue)
hasLineNum	이차원 모양에 포함된 선의 수
cx, cy	이차원 모양의 중심점
mX,mY, miX,miY	이차원 모양을 포함하는 최소 사각형의 범위를 알려주는 값
size	이차원 모양의 크기
isInferred	이차원 모양의 추론 여부
isClosed	이차원 모양의 개폐(開閉) 여부

[표 2]는 [표 1]에서 소개한 정보들을 이용하여 [그림 4]에서 주어진 사각형(tetragon) 모양에 대한 속성 값을 나타낸다.

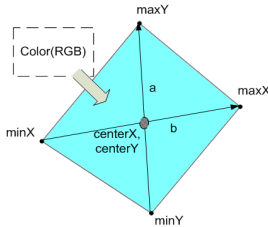


그림 4. 사각형 모양의 속성 값

표 2. Lv2\_01 속성의 수치화 표현

ID	Lv2_01
type	tetragon
hasLineNum	4
size	$(maxY - minY) * (maxX - minX)$
isInferred	false
isClosed	true

### 3.3.1 이차원 모양의 유형 정의

본 논문에서는 모양들의 유형을 크게 연결선(lines), 정규모양(regular shape), 비정규모양(irregular shape) 들로 나눈다. 연결선은 내부 영역을 갖지 않은 단순한 선들의 연결이다. 정규 모양은 삼각형(triangle), 사각형(tetragon), 원(circle), 타원(ellipse)으로 분류하고 그 외

의 모양들은 모두 비정규 모양으로 분류한다. 인식 단계에서 정의한 객체의 모양을 나타내는 기본 단위(primitive unit)는 삼차원 모양들이다. 삼차원 기본 모양들은 상자(box), 원통(cylinder), 구(sphere), 피라미드(pyramid)등으로 정의한다. 에이전트는 여러 장면에서 지각되어진 이차원 모양의 정보들을 조합하여 삼차원 객체를 인식한다. 본 논문에서는 정규 모양의 종류를 최소화 하는 이유는 에이전트의 시야 각도에 따라서 정사각형 모양이 이등변 삼각형 모양으로 보이기도 하고 정사각형 모양이 직사각형 또는 마름모 모양으로 보이는 경우가 많기 때문이다. 하지만 원은 삼각형이나 사각형 모양처럼 타원 모양으로 일반화하지 않는다. 왜냐하면 어떤 객체가 연속적인 장면에서 모두 원으로 지각된다면 이차원 모양이 삼차원 구 모양이라는 것을 의미하기 때문이다. 본 논문에서는 비정규형 모양들도 정규모양들을 이용하여 근사화 할 수 있다. 즉, 일부 부분이 손실된 모양이나 둘 이상의 도형이 합쳐져 복합적인 모양들은 정규 모양들의 조합으로 표현할 수 있다[17].

### 3.3.2 이차원 모양의 유형 추론

본 논문에서는 부분적인 손실을 가진 모양들을 모두 추론 모양(Inferred Shape)으로 정의한다. 예를 들어, 세 선들이 닫혀있는 것이 삼각형이지만 사람은 삼각형의 짧은 구간이 끊어져있거나 다른 사물들에 의해 삼각형의 일부가 가려져 있더라도 나머지 부분들을 통해 그것을 삼각형으로 추론 할 수 있다. 그리고 모양의 유형 결정 단계에서는 추론 모양들을 따로 분류하여 추론 알고리즘을 적용한다. 추론 알고리즘을 적용한 결과로 비정규 모양이 정규모양의 일종(一種)으로 판정할 수 있다. [그림 5]는 추론 알고리즘을 적용하여 손실된 모양들을 보여주는 그림이다.

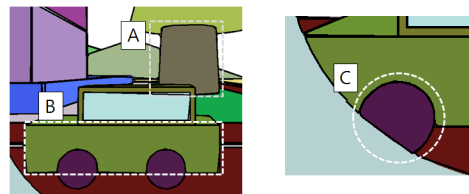


그림 5. 추론 알고리즘을 적용하는 손실 모양들

가상 세계에서 모양의 손실은 다른 객체에 가려짐으로써 발생할 수도 있고 에이전트 시야의 불완전성에 의해서도 발생한다. 추론 알고리즘은 손실된 일부분을 제외한 나머지 부분을 이용하여 원래의 모습을 복원하는 알고리즘이다. 추론 알고리즘은 세 가지로 나눌 수 있다. 첫 번째로는 [그림 5]에서 모양 A처럼 두 개의 연장선을 계산하고 그 둘의 교점을 기준으로 나머지 부분을 추론하는 방법이다. 두 번째로는 모양 B처럼 하나의 선이 외부 요인에 의해 부분적으로 끊어졌다고 가정하고 그 선을 추론하는 방법이다. 마지막으로 모양 C처럼 타원 모양과 같은 형태의 경우에는 반복되는 각도와 선의 길이를 먼저 계산하고 각도와 길이만큼 새로운 선들을 연결하여 손실된 부분을 추론하는 방법이다. 어떤 추론 방법을 적용할 것인지 결정하는 요소는 각 선들이 이루는 각도(모양 내부를 향해 시계 방향으로 측정한 각도)와 겹쳐진 선(folded line)들(다른 모양이나 에이전트 시야의 가려짐으로써 만들어진 선들)이다. [그림 6]은 모양 A에서 선들이 가진 각도들과 겹쳐진 선들을 표시한 그림이다.

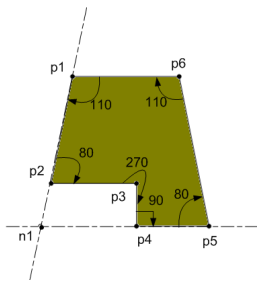


그림 6. 모양 A의 추론

두 선이 이루는 각도는 모양 내부 방향으로 측정한다. 겹쳐진 선들은 두 선들이 이루는 각도가 180도를 초과한 선들을 말한다. [그림 6]에서 Line p2-p3와 Line p3-p4는 270도의 각도를 가지므로 겹쳐진 선으로 간주된다. 이런 겹쳐진 선들과 연결된 선들을 가상 선(imaging line)들이라고 정한다. 추론 알고리즘은 손실 모양의 선들이 이루는 각도와 가상 선들을 조건으로 [표 3]의 추론 알고리즘 가운데 하나를 손실된 부분에 적용한다. [그림 6]의 모양 A는 조건 1에 의해서 새로

운 점과 선들이 생기고 이 성분들에 의해서 새로운 속성 값들을 갖게 된다.

표 3. 모양 추론 알고리즘 종류

종류	적용 조건	변경 사항
모서리 손실 추론	가상 선들이 하나의 교점을 갖는 경우	1.교점에 의해 생긴 새로운 두 선을 모양에 추가시킴. 2.정규모양 유형과 모양의 크기, maxX, maxY, minX, minY값을 다시 계산하여 저장함.
선 손실 추론	손실 모양의 가상 선들이 같은 선상에 존재하는 경우	1.가상 선들을 모두 포함하는 하나의 선을 새로운 선으로 저장 2.정규 모양의 유형과 size, maxX, maxY, minX, minY값을 계산하여 저장함.
타원 추론	손실 모양의 선들이 가진 각도가 오차 범위 내의 값들로 반복되고, 선의 길이가 오차 범위 내에서 일정한 경우	1.반복되는 각도만큼 새로운 선들을 계속 연결하여 타원 형태를 구성. 2.정규 모양 유형과 모양의 size, maxX, maxY, minX, minY값을 다시 계산하여 저장함.

[그림 7]은 [그림 5]의 모양 B를 추론하기 위해 점과 선들로 표현한 그림이다. 모양 B에서 점 p4 ~ p8 그리고 p13 ~ p17의 각도는 180°이상이다. 따라서 Line p2-p3, Line p10-p11, Line p18-p19가 각각 가상 선들이 된다. 이 선들은 Line p2-p19 위에 존재하므로 모양 B에 조건 2를 적용한다.

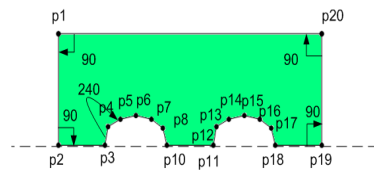


그림 7. 모양 B의 추론

아래 [그림 8]은 [그림 5]의 모양 C를 표현한 그림이다. 원 모양은 중점을 기준으로 일정한 길이와 각도들이 선들로 연결된 것이다, 이 원리를 이용하여 나머지 선들을 연장하는 방법을 사용한다. 모양 C는 조건 3에 의해 새로운 선들이 생성되며 추론된 원의 속성 값들을 저장한다.

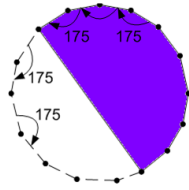


그림 8. shape C의 추론

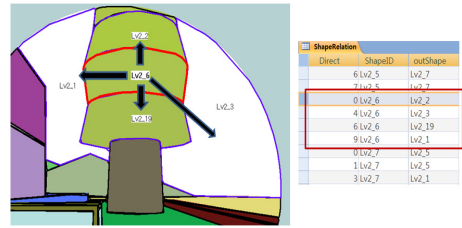


그림 9. 위치 관계의 표현

### 3.4 이차원 모양들 간의 공간 관계

#### 3.4.1 외부적 위치 관계(touch relations)

일반적으로 모양들은 다른 모양들과 다양한 공간적 위치 관계를 갖는다. 인식에서 이차원 모양 간의 위치 관계는 삼차원 모양을 만들거나 삼차원 모양 간의 위치 관계를 파악하는데 이용된다. [표 4]의 shapeRelation 테이블은 모양들의 위치 정보를 저장한다.

표 4. ShapeRelation 테이블

Field	설명
SceneNum	이 정보가 어떤 장면에 속해 있는지 알려주는 값
ShapeID	어느 이차원 모양의 위치 관계 정보인지 알려주는 값
Direction	해당 이차원 모양의 몇시 방향에서 위치 관계가 발생하였는지 알려주는 값
touchShape	현재 방향에서 위치 관계를 갖는 외부 모양의 ID
hostShape	현재 방향에서 자신을 포함하고 있는 외부 이차원 모양의 ID
includeShape	현재 방향에서 자신이 포함하고 있는 내부 이차원 모양의 ID

위치에 관한 정보는 [표 4]에서 보는 바와 같이 ShapeRelation 테이블의 Direction, touchShape 정보와 밀접한 관계를 갖는다. touchShape는 모양들을 0시 ~ 11시 방향 까지 12개의 방향에 대해 다른 모양들과 위치 관계를 검사받으며 이때 발견된 모양의 ID를 저장한다. Direction은 위치 관계가 발생한 두 모양의 중점이 이루는 각도를 기준으로 결정된다. [그림 9]는 Lv2.6 모양이 다른 모양들과 갖는 위치 관계와 그것이 어떤 형태로 shapeRelation 테이블에 저장되는지를 보여주고 있다.

#### 3.4.2 포함 관계(In and Include relations)

본 논문에서는 외부적 위치 관계와 더불어 이차원 모양들 사이의 포함 관계를 'In'과 'Include'로 표현한다. 예를 들어 사각형(tetragon) 모양 A의 내부에 원(circle) 모양 B가 위치할 경우 A는 B를 'Include'라 하고 B는 A에 'In'되어 있다고 정의한다. 두 가지 표현을 중복 사용한 이유는 모양의 최외곽에서 'Include' 하는 모양을 찾거나 가장 내부의 'In'하는 모양을 찾을 때 좀 더 빠르게 DB를 검색하기 위함이다. 모양이 다른 경우 모양들의 'in'과 'include' 관계는 우리 주위에서 흔히 볼 수 있는 위치 관계이다. 예를 들어 자동차의 창, 얼굴 안의 이목구비, 문의 손잡이 등의 객체들도 'In'과 'Include' 관계를 가진 객체들이다.

'In'과 'Include' 관계 표현은 [표 4]의 Direction, hostShape, includeShape 정보를 사용하여 표현한다. 아래 [그림 10]은 다양한 'In'과 'Include' 상황이 복합적으로 나타난 경우를 보여주고 있다. 그리고 아래 [표 5]는 [그림 10]의 관계들을 표현한 결과이다. A가 C를 'Include'하고 C가 D를 'Include' 하고 있으므로 재귀적인 방법(recursive)을 적용하면 A가 D를 'include'하고 있다는 것을 알 수 있다.

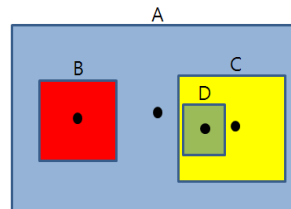


그림 10. 모양들 간의 포함 관계

표 5. 'In'과 'Include' 관계를 가진 복합적인 모양들의 표현

Direction	ShapelD	host Shape	Include Shape
3	A	null	C
9	A	null	B
3	B	A	null
9	C	A	D
3	D	C	null

#### 4. 지각 데이터의 활용

##### 4.1 지각 데이터를 이용한 객체 인식

에이전트가 지각된 여러 가지 요소들을 조합하여 특정한 객체로 판단하는 것은 기억 속에 그 객체에 관한 정보를 가지고 있기 때문이다[13]. 본 논문에서는 객체들의 여러 속성들 가운데 부분(part)의 유형(type)과 부분들 사이의 위치 관계 그리고 색등의 속성들을 이용하여 객체들을 규정한다. [그림 11]은 자동차를 여러 각도에서 지각한 화면들을 보여주고 있다.

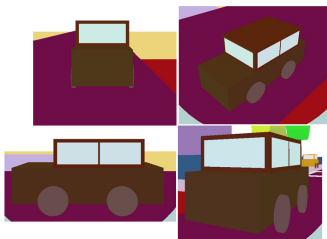


그림 11. 여러 각도에서 지각된 삼차원 자동차

이때 삼차원 이미지를 인식하기 위해서는 [그림 11]에서처럼 객체를 어느 각도에서 지각하더라도 지각된 정보를 바탕으로 동일한 객체로 인식할 수 있어야 한다. 본 논문에서 에이전트는 객체를 다를 각도에서 보더라도 동일한 객체로 인식할 수 있어야 한다. 따라서 객체의 정의는 다양한 각도에서 지각된 정보를 수용할 수 있도록 구체화한다. 아래 [그림 12]는 자동차 모양의 정의를 도식화한 것이다. 자동차의 구조를 삼차원적으로 표현하기 위해 상단(Cabin) 부분과 몸체(Body) 부분으로 나누고 자동차의 구성 요소인 창문(Window)과 바퀴(Wheel)의 위치 관계를 인식 요소로 이용한다.

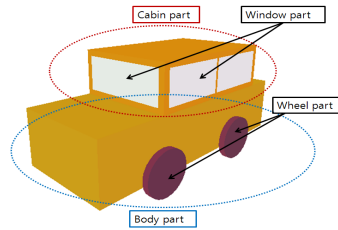


그림 12. 자동차의 정의

인식은 3장에서 소개했던 이차원 모양의 유형, 색, 크기, 이차원 모양들 간의 위치 관계를 이용하여 삼차원 기본 모양을 구성하는 것으로 시작한다. 이의 결과로 도출된 삼차원 기본 모양들(3D primitive shapes)을 이용하여 자동차의 속성 표현을 나타낸다. [표 6]은 자동차의 속성 표현을 구체적인 수치로 나타낸 것이다.

표 6. 자동차의 정의

Part	삼차원 모양	color (범위)	다른 부분와의 connection(연결 type: direction범위)
Cabin	Box	all color	Window(Include:all direction) Body(Touch:4시~8시)
Window	Tetragon	R,G,B모두 (200~250)	Cabin(In:all direction)
Body	Box	all color	Cabin(Touch:10시~2시) Wheel(Touch:4시~8시)
Wheel	Cylinder	all color	Body(Touch:10시~2시)

삼차원 기본 모양들이 [표 6]의 정보 중 두 가지 연결 조건을 만족하면 객체를 자동차로 인식한다. 각 객체의 특징을 명확히 규정짓는 값들은 가상 세계를 구현하는 개발자에 의해서 목적에 맞게 정한다. 본 논문에서 사용된 객체들은 자동차와 마찬가지로 기본 모양들, 색, 부분, 부분들 간의 연결을 기본 요소로 정의한다.

##### 4.2 객체의 시공간적 위치 표현

Spatio-Temporal Graph 는 각각의 객체들이 가상 세계에서 차지하는 시공간적 위치를 표시한다. [그림 13]은 가상 상황의 예제 전경(全景)에 상응하는 S-T 그래프의 내용을 나타낸다.



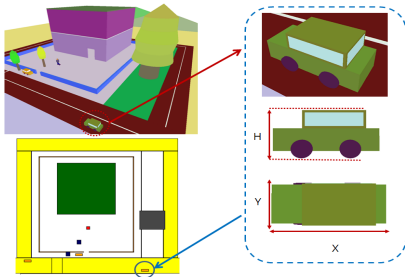


그림 13. 시각 데이터를 이용한 S-T 그래프 상의 객체 표현

X, Y 좌표는 객체가 그래프 상에서 차지하는 평면적 영역을 표현하였고 객체의 중점을 기준으로 S-T 그래프 상에 위치시켰다. H는 객체의 높이를 나타내고 그래프 상에서 색의 차이를 이용하여 높낮이를 표현 한다 [18].

S-T 그래프 상의 내용은 주기적으로 갱신된다. 아래 [그림 14]은 세 장의 S-T 그래프들을 3초의 간격으로 보여주고 있다. 각 상황의 발생 시간은 SceneTime 테이블로 부터 가져온다. 이 S-T 그래프들을 분석하면 물체의 이동 경로를 파악하고 더 나아가 미래의 위치를 예측하는 일이 가능하다.

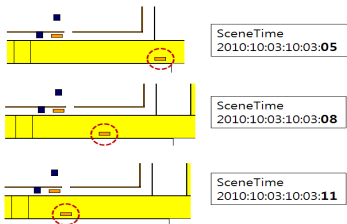


그림 14. S-T 그래프에 표현된 상황의 변동

에이전트는 S-T 그래프를 지식 모델링에 이용할 수 있다. 예를 들어 에이전트 과거의 어느 시점에 어떤 객체를 보았다면 현재에 그 객체가 어디에 있는지를 기억해 낼 수 있다. [그림 15]는 [그림 14]에 표시된 자동차의 이동을 나타낸 DB데이터이다.

sceneNum	UnitID	uHeight	uCenterX	uCenterY	uVolumeX	uVolumeY
51	PassCar	75	2300	100	100	40
52	PassCar	75	1300	100	100	40
53	PassCar	75	300	100	100	40

그림 15. PassCar의 이동 경로

51번 장면부터 53번 장면까지 3초간 PassCar는 1000만큼씩 동쪽에서 서쪽으로 이동하고 있다. 이것을 근거로 54번 장면에서는 PassCar가 -700지점에 있을 것이라고 예상할 수 있다.

### 5. 구현 및 실험 결과

본 논문에서의 시뮬레이션은 VC++, Open GL, 그리고 Microsoft Access 2007 데이터베이스를 이용한다. 앞에서 언급된 이론들을 적용하여 에이전트가 본적 없는 마을의 풍경을 인식하면서 기억해 나가는 것을 구현한다. 시뮬레이션에서 등장하는 객체는 에이전트와 함께 자동차, 나무, 담장, 건물, 계단 등이며 객체들의 정의는 4.1장에서 언급했던 것처럼 기본 모양, 색, 부분들의 유형, 부분들의 연결들을 사용한다. 시뮬레이션에서 제공하는 사용자 인터페이스는 세 부분으로 구성된다. [그림 16]은 시뮬레이션에서 사용자 인터페이스를 보여주는 화면이다.

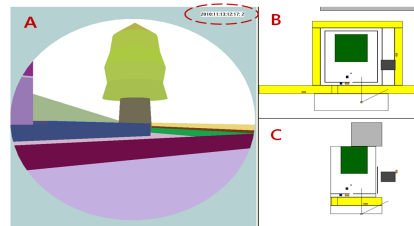


그림 16. 시뮬레이션에서 사용자 인터페이스 소개

[그림 16]에서 사용자가 주인공 에이전트의 시야를 쫓아가며 어떤 데이터들을 시각하는지 보여주는 화면 (A), 가상 세계의 시공간 정보를 담고 있는 전지적인 (omniscient) Global S-T 그래프 화면(B) 그리고 각 에이전트 개인들의 지식 내용을 표현한 에이전트의 지식 그래프 화면(C)로 구성된다.

시뮬레이션에서 등장하는 주인공 에이전트에게는 Tom이라고 ID를 부여한다. Tom은 가상 마을에 처음 방문 하였으며 마을에 들어서는 순간부터 주위 환경 정보를 시각한다. 그리고 Tom은 그 정보를 바탕으로 객체들을 인식하면서 동시에 그들의 위치를 계산하여 자신의 지식의 표현 방법 인 S-T 그래프에 저장시키는

과정을 반복한다. [그림 16]에서 A부분은 에이전트 Tom의 시야 범위와 시야 안의 풍경을 보여주는 화면이다. B부분은 Global S-T 그래프를 나타내며 C부분은 에이전트의 지식속의 그래프이다. 사용자는 마우스를 이용하여 화면을 축소하거나 확대 할 수 있다. B와 C 영역은 아래 부분에서 상세히 설명한다.

[그림 17]은 [그림 16]의 Global S-T 그래프를 확대한 그림이다.

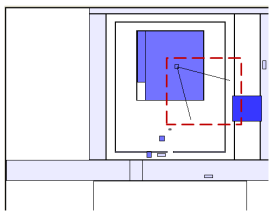


그림 17. Global S-T 그래프

점선 박스는 Tom의 위치를 나타내며 Tom의 시야 방향과 범위는 두개의 직선을 이용하여 표시한다. 이 그래프는 마을 전체를 공간적으로 구성한 모습과 움직이는 객체들이 시간에 따라 객체가 점유하고 있는 위치, 범위, 이동 그리고 물리 현상을 추론할 수 있는 시공간 정보를 나타낸다.

[그림 18]은 [그림 16]의 C화면을 확대한 그림이다.

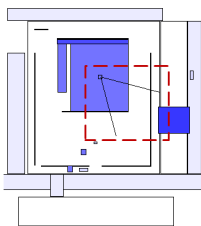


그림 18. Tom의 지식 속의 그래프

Tom은 마을에 처음 방문했으므로 초기에는 마을에 대한 정보가 전혀 없지만 마을 이곳저곳을 다니면서 정보량이 늘어나고 궁극적으로 Global S-T 그래프와 유사한 모양으로 객체들과 그들의 위치를 기억하게 된다.

[그림 19]는 10분 동안 Tom이 마을의 이곳저곳을 둘러보면서 인식한 정보들을 자신의 지식 속의 그래프에

표현한 화면들을 연속적으로 보여주는 그림으로 마을에 대한 Tom의 정보량이 점차 늘어나는 것을 알 수 있다.

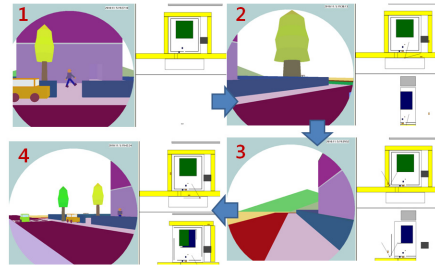


그림 19. 2분 동안 Tom이 인식한 마을의 정보를 연속적으로 표현한 그림

Tom은 [그림 19]에서 4번 장면 전까지 멀리 떨어져 있는 산의 위치나 건물의 높이 등을 정확히 알지 못했기 때문에 자신이 갖고 있는 지식 속의 산과 빌딩에 관한 표준적 기본 정보를 이용하여 객체들을 인식하고 위치를 정한다. 또한, Tom은 새로운 객체를 인식하는 작업과 함께 기존 객체의 정보와 자신이 이동하면서 얻은 새로운 객체 정보를 더해서 잘못된 정보들을 수정해 나가는 것도 볼 수 있다.

[그림 20]은 [그림 19]의 1번 장면과 4번 장면의 산과 건물 정보를 나타낸 데이터 정보이다.

Instance	StructureUnits	sceneNum	type	ID	x1	y1	x2	y2	height
2	Car	brownCar	-300	-440	-100	-360	75		
3	road	road3	2100	-3700	2120	-300	3		
3	mount	mt2	200	-3900	2120	-4700	2000		
3	roof	Fl2_roof	200	-3000	800	-1680	711		
3	mount	mt1	-350	-4250	450	-5700	2000		
157	wall	Fl2_dw1	-692	-1700	800	-1680	1220		
157	mount	mt1	-2251	-6142	450	-5700	2000		
157	mount	mt2	-1492	-5192	3307	-4700	1800		
157	roof	Fl2_roof	-683	-3277	800	-1680	1240		

그림 20. 장면 65와 장면 157의 객체 정보

Tom의 지식 속에 일반적인 산의 높이는 2000 가량이 다. 따라서 정확한 산의 높이를 알 수 없을 때 sceneNum 3에서 두 산 mt1과 mt2의 높이가 각각 2000인 이유가 바로 이것 때문이다. 장면의 정보는 시간이 흐름에 따라 점점 변한다. Tom은 장면 157에서 Global S-T 그래프의 mt1과 mt2의 높이인 2000, 1800과 같은 값으로 산들의 높이를 인식했다는 것을 알 수 있다.

[그림 21]은 에이전트의 위치에 따라 에이전트의 지식상의 건물 내부 모습이 달라지는 것을 보여주는 그림이다.

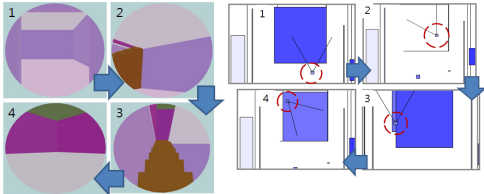


그림 21. 에이전트의 이동에 따른 건물 내부 표현의 변화

내부의 모습을 다르게 만든 이유는 2.5차원으로 지식 속의 그래프가 구성되었기 때문이다. 건물 내부의 정보를 사용자 인터페이스에 효과적으로 보여주기 위해서 Tom이 현재 위치하고 있는 층의 정보만 에이전트의 지식 속의 그래프에 표현한다.

[그림 22]는 [그림 21]의 1번(장면 65)과 4번(장면 92)순간에 Tom이 지각한 객체 데이터를 보여주는 그림이다.

Instance				Instance			
sceneNum	type	ID	x1	sceneNum	type	ID	x1
65	road	road2	-1900	92	fence	buf_dw1	-1050
65	crosswalk	crosswalk	-950	92	stairs	stairs1	-780
65	mount	mt1	-2251	92	flatlad	centerScetion	-1442
65	fence	buf_dw3	-1300	92	wall	fl1_dw4	780
65	Car	AceCar	0	92	road	road4	-3150
65	fence	buf_dw4	1270	92	wall	fl2_dw1	-704
65	road	road1	-1792	92	roof	fl1_roof	-592
65	fence	buf_dw5	-1300	92	tree	giantTree	1450
65	wall	fl1_dw5	-800	92	tree	tree1	-260
65	wall	fl1_dw4	780	92	mount	mt2	-1492
65	fence	buf_dw1	-1050	92	Car	PassCar	-1000
65	wall	fl1_dw1	0	92	fence	buf_dw2	100
65	wall	fl2_dw5	-800	92	tree	tree2	-560
65	tree	giantTree	1450	92	road	road3	2100
65	tree	tree2	-560	92	flatlad	treeScetion	1500
65	roof	fl1_roof	-592	92	flatlad	start_section	-1350
65	Car	brownCar	-300	92	roof	fl2_roof	-650
65	tree	tree1	-260	92	wall	fl1_dw1	0

그림 22. 장면 65와 장면 92의 객체 데이터 비교

장면 65까지 Tom이 지각한 객체 정보에 계단이 없지만 장면 92에는 계단(stairs:ID stairs1)의 높이와 위치 정보가 저장되어 있다는 것을 확인할 수 있다.

여기서 주목할 점은 건물 내부로 들어간 후의 Tom과 그 이전 Tom의 객체 정보 차이이다. 만약 장면 65의 순간에 사용자가 Tom에게 “건물의 2층으로 가는 방법을 알고 있습니까?”라고 질문 했을 때 Tom은 모른다고 대

답했겠지만 장면 92 이후에 그 질문을 받았을 때는 “계단을 이용해서 올라갈 수 있다.”라고 대답할 수 있다. 이런 차이를 보이는 이유는 Tom이 계단에 관한 정보와 더불어 장면 66부터 장면 92까지의 정보들을 기억하고 있기 때문이다. 이처럼 시간의 흐름에 따른 정보의 차이로 인해 Tom의 대답이 달라지고 사용자는 Tom의 대답을 근거로 그의 과거 경험을 예측하는 동시에 새로운 정보를 얻으면서 가상 세계에서의 몰입감을 유지해 나갈 수 있다.

## 6. 결론 및 향후 연구 방향

본 논문은 에이전트를 실제 사람과 유사하게 행동하도록 만들기 위해 지각과 인식 기능 그리고 에이전트의 시공간 지식을 위한 지식 표현 방법을 개발하였다. 본 논문의 핵심 기술들은 이차원 화면을 통해서 삼차원 객체의 모습을 인식할 수 있는 알고리즘과 손실되거나 가려진 경우에도 모양을 추론할 수 있는 알고리즘이다. 사람의 행동은 선천적으로 타고난 기질과 후천적으로 익힌 학습에 의해 결정 되는 것처럼 가상 세계의 에이전트도 탄생 순간에 부여된 성격의 속성 값과 가상 세계에서 활동하면서 얻은 정보량에 의해 행동 성향이 결정된다. 본 논문은 객체를 인식하고 그들의 위치를 장면 단위로 기억하면서 좀 더 긴 시간동안 넓은 범위를 다니면서 다양한 상황을 지각한 에이전트가 더 많은 정보를 가질 수 있도록 설계하였다. 그러나 실세계와 유사한 가상 환경의 지각에 적용하기 위해서는 좀 더 다양한 지각 알고리즘이 필요하다. 여기에서 소개한 시물레이션은 가상 환경을 극단적으로 추상화 시킨 예제이다. 그렇기 때문에 시물레이션에 등장하는 모양의 유형과 그것들의 위치 관계 그리고 부피를 효과적으로 지각하면서 각각의 객체를 인식하는데 큰 문제를 발생시키지 않았다. 하지만 실제로 사용자에게 몰입감 높은 세상을 보여주기 위해서는 복잡하게 구성된 객체들이나 다양한 형태를 가진 객체를 지각할 수 있는 알고리즘을 추가시켜야만 한다. 이것을 위해서는 지각 자체의 알고리즘 개발뿐만 아니라 인식 단계의 객체 구성과 이벤트 분석 알고리즘을 파악하고 그것에 알맞은 형태의 데이터를 추출하는 기술 개발에도 힘써야 한다.

## 참고 문헌

- [1] J. H. Park, "A logical simulation of discretionary events in spatio-temporal context," Tech report #30, AIMM lab, Kyungpook Nat'l Univ, 2004.
- [2] 황원택, *Prop의 행동 다양화를 위한 인지중심방식 접근 및 FSM응용모델*, 경북대학교 석사학위논문, 2001.
- [3] J. E. Lird, "Using a computer game to develop advanced AI," Computer, Vol.34, Issue.7, pp.70-75, IEEE, 2001(7).
- [4] 정근재, "가상세계에서의 온톨로지 기반 객체 인식 기법", 한국콘텐츠학회논문지, 제9권, 제4호, pp45-54, 2009.
- [5] 지세진, "가상 주민의 인간 같은 행동을 시뮬레이션 하기 위한 지식 모델", 경북대학교 박사학위논문, 2007.
- [6] 이상봉, *이차원 그래픽 정보의 세만틱 표현방법의 개발*, 경북대학교 석사학위 논문, 1993.
- [7] J. Song, M. Cai, M. Ly, and S. Cai, "Graphics recognition form binary iamges: One step or two steps," International Conference on Pattern Recognition(ICPR), Vol.3, pp.135-138, 2002.
- [8] D. Dori, "Sparse pixel vectorisation: An algorithm and its performance evaluation," Pattern Analysis and Machine Intelligence (PAMI), Vol.21, pp.202-215, 1999.
- [9] S. Yoon, G. Kim, Y. Choi, and Y. Lee, "New paradigm for segmentation and recognition," Workshop on Graphics Recognition(GREC), pp.216-225, 2001.
- [10] J. Song, F. Su, C. Tai, and S. Cai, "An object-oriented progressive-simplification based vectorization system for engineering drawings: Model, algorithm and performance," Pattern Analysis and Machine Intelligence(PAMI), Vol.24, pp.1048-1060, 2002.
- [11] 마이크로소프트웨어. *시맨틱 웹의 핵심 기술*, 온톨로지, 2002(4).
- [12] B. Chandrasekaran and John R. Josephson, "What are ontologies, and why do we need them?," IEEE Intelligent Systems, 1999.
- [13] 김동훈, *온톨로지의 효율적인 구성을 위한 스키마 통합 기법의 개발 및 구현*, 경북대학교 석사학위 논문, 2008.
- [14] S. Kim, H. Kim, J. S, and I. S. Kweon, *지능형 이동 로봇에서 강인 물체 인식을 위한 영상 문맥 정보 활용 기법*, 로봇공학회연구지, 2006.
- [15] S. Ullman, *High-Level Vision*, MIT Press, 1996.
- [16] C. O. Yun, T. S. Yun, and D. H. Lee, "Interactive 3D Stereoscopic Image Editing System using Image-based modeling," KSIAM IT series Vol.10, No.2, pp.55-66, 2006.
- [17] J. H. Park. and B. s. Um, "A new approach to similarity retrieval of 2-D graphic objects based on dominantshapes," Pattern Recognition Letters 20, pp.591-616, 1998.
- [18] 조규명, *Spatio-temporal graph를 이용한 가상 현실 속의 상황 표현 방법 모델링*, 경북대학교 석사학위 논문, 2010.

## 저자 소개

박재우(Jae-Woo Park)

준회원



- 2006년 2월 : 영남대학교 전자공학과(공학사)
- 2011년 2월 : 경북대학교 전자전기컴퓨터학부(공학석사)

<관심분야> : 지각, 인식, 온톨로지

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 중 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