

지능형 로봇 비전 프레임워크: VisionNEO

장세인* · 박충식* · 우영운** · 김광백***

*영동대학교 컴퓨터공학과

**동의대학교 멀티미디어공학과

***신라대학교 컴퓨터정보공학부

An Intelligent Robot Vision Framework

Se-In Jang* · Choong-Shik Park* · Young-Woon Woo** · Kwang-Baek Kim***

*Department of Computer Engineering, Youngdong University

**Department of Multimedia Engineering, Dong-Eui University

***Division of Computer and Information Engineering, Silla University

E-mail : 75100113106s@gmail.com, leciel@youngdong.ac.kr, ywwoo@deu.ac.kr, gbkim@silla.ac.kr

요 약

오늘날 지능형 로봇은 국, 내외로 많은 관심을 받고 있는 분야이다. 지능형 로봇이란 외부환경을 인식하고 스스로 판단하여 자율적으로 동작을 하는 로봇을 의미한다. 이에 대한 연구 개발이 활성화됨에 따라 로봇 소프트웨어 개발을 효과적으로 지원하기 위한 로봇 소프트웨어 플랫폼에 대한 연구가 활발해지고 있다. 시시각각 변화하는 환경에서 민감하게 반응하기 위해서는 시각센서를 이용하여야 하고, 자신의 행위를 적절히 대응시키기 위해서는 주변 상황과 알맞은 행동을 추론하고 학습해야 한다.

본 연구에서는 인공지능 규칙처리 추론엔진을 토대로 한 NEO 시스템에 영상 처리 시스템을 올려 지능형 로봇을 제어하는 루틴을 추가한 VisionNEO를 개발하였다. 그리하여 주변 환경을 이해하고 알맞은 행동을 추론, 학습해 지식을 축적하는 지능형 로봇 비전 프레임워크를 제안한다.

키워드

Intelligent, Robot, Vision, Framework, NEO, Embodied Cognition

I. 서 론

오늘날 지능형 로봇은 국, 내외로 많은 관심을 받고 있는 분야이다[1]. 지능형 로봇이란 외부환경을 인식하고 스스로 판단하여 자율적으로 동작을 하는 로봇을 의미한다. 시시각각 변화하는 환경에서 민감하게 반응하고 자신의 행위를 적절히 대응시키기 위해서는 주변 상황과 알맞은 행동을 추론해야 한다. 또한 다양한 목적을 수행하면서 이전에 했던 실수를 반복하지 않기 위해서는 학습 기능을 가지고 있어야 한다. 이러한 시스템을 만들기 위해서는 여러 기술들을 효과적으로 통합해야 할 것이다. 하지만 주변 환경을 이해하고 효과적

으로 추론해 낼 수 있는 시스템은 그 개념이 명확하게 정립되지 않아 시스템 구현이 어렵다.

주변 환경의 변화를 획득할 수 있는 감각능력 중 하나인 시각센서는 가장 민감하게 많은 변화를 찾아낼 수 있으며 사물과 직접 접촉하지 않고도 정보를 얻을 수 있는 강점이 있다. 시각센서를 이용한 컴퓨터 비전 분야는 오래전부터 영상에서 장면이나 특징들을 이해하는 것을 목적으로 하고 있으며 영상 처리에 대한 시스템 연구가 활발히 진행 되고 있다[2].

본 연구에서는 인공지능 규칙처리 추론엔진을 토대로 한 NEO 시스템에 영상 처리 시스템을 올려 VisionNEO를 개발한다. 그리하여 주변 환경을 이해하고 알맞은 행동을 추론, 학습해 지식을 축적하는 지능형 로봇 비전 프레임워크를 제안한다.

* 본 논문은 2008년 정부(교육과학기술부, 한국산업기술재단)의 지원으로 영동대학교 공학교육혁신센터지원사업의 연구지원을 받아 수행된 연구입니다.

II. 관련 연구

지능형 로봇에 대한 연구 개발이 활성화됨에 따라 로봇 소프트웨어 개발을 효과적으로 지원하기 위한 로봇 소프트웨어 플랫폼에 대한 연구가 활발해지고 있다. Microsoft사의 MSRS(Microsoft Robotics Studio), Evolution Robotics사의 ERSP(Evolution Robotics Software Platform), 유럽에서 오픈소스 프로젝트로 시작된 ORCOS(Open Robot Control Software), Interbots사의 BAT(Behavior Authoring Tool), NEC사의 Robostudio 등 다양한 플랫폼이 존재한다[3]. 소개된 플랫폼들은 로봇 응용을 손쉽게 개발하기 위해 나름대로의 기능을 제공하고 있다. 하지만 로봇이 스스로 추론할 수 있다거나 학습할 수 있는 부분이 미비하다고 할 수 있다.

컴퓨터 비전 분야에서 장면이나 특징을 이해하기 위해서는 수학적 기초지식을 갖추고 그에 대한 기법을 실제 영상에 적용하는 과정이 필요하다. 이러한 과정을 이해하고 나서야 영상처리가 가능했지만 그 과정을 줄여 용이하게 접근하도록 도와주는 매개체가 존재한다. 영상처리 라이브러리는 Matrox-MIL, OpenCV, CxImage 등이 존재한다. 이외에도 비주얼 프로그래밍 도구(Visual Programming Tool)를 기반으로 한 영상 처리 시스템으로 Ad-Oculus, NeatVision, Matlab, Khoros가 존재한다[4]. 하지만 영상 처리 라이브러리의 경우는 그에 대한 구조를 익히고 프로그래밍 해야만 가능하고, 비주얼 프로그래밍 도구의 경우는 개발용으로는 부적합하다.

위의 연구들로 보아 영상처리에 대한 학습과정과 실제 응용 시스템 개발과정이 나뉘어 있고, 시뮬레이션 및 개발을 위한 환경으로 그 사용법이 복잡한 특징을 가지고 있다.

III. VisionNEO

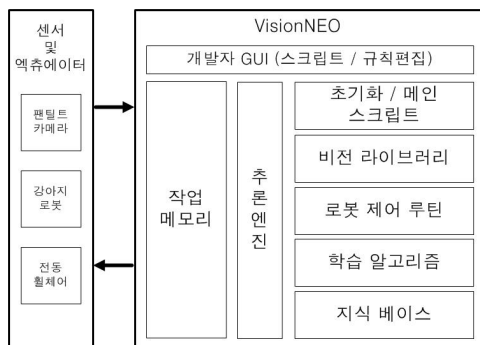


그림 1. VisionNEO의 구조

VisionNEO는 인공지능 규칙처리 추론엔진을 토대로 한 NEO(Name Equals Object) 시스템과 비전 처리 시스템, 지능형 로봇을 결합한 로봇 소프트웨어 플랫폼이다. 또한 IAS(Intelligent Agent System)를 이용하여 각 에이전트끼리 정보를

전달하기 위해 이루어져있고 지능형 로봇과 비전 처리 시스템을 NEO에서 간단한 명령어로 사용할 수 있도록 구현한다.

3.1 NEO(Name Equals Object) 시스템

NEO 시스템은 인공지능 프로그램을 개발하기 위해 자체 개발되었다. NEO는 스크립트 기반의 인공지능 규칙처리 추론엔진을 토대로 한 시스템이다. NEO에서는 모든 개체는 리스트로 이루어져 있다. 리스트는 문자나 숫자로 이루어진 구성요소를 나열한 것이며, 그렇게 이루어진 구성요소도 또 다른 리스트의 구성요소가 될 수 있다[5].

NEO는 함수 언어 기능, 객체지향 기능, 규칙추론 기능 등을 지원하지만 이러한 기능을 이루는 함수, 객체, 속성, 메서드, 사실, 규칙 등 모든 구성요소가 리스트 형태로 표현되고, 관리된다. 모든 구성요소가 리스트로 구성, 관리되는 이유는 시스템의 모든 자원을 투명하고 일관성 있게 처리하기 위함이다. NEO의 수행방식이나 함수 언어 기능은 LISP와 유사하지만, 내부적으로 처리되는 방식과 내장 함수들은 서로 다르며, 객체지향 기능이나 규칙추론 기능, 진리치 유지 기능은 지식기반 시스템 개발을 위한 환경이 된다.

NEO의 핵심 기능은 DLL형태로 이루어져 있기 때문에 인공지능 프로그램을 개발할 때 내장하여 이용할 수 있으며, GUI를 Visual Basic, Visual C++를 이용하여 구현할 수 있다. NEO의 개발환경은 Windows 95/NT에서 수행되는 명령어 인터프리터 형태로 되어있는데 이 또한 NEO의 DLL을 이용하여 만들어진 것이다.

3.2 NEO IAS(Intelligent Agent System)

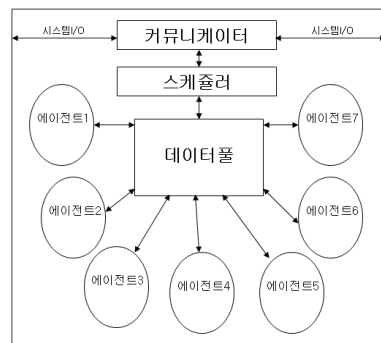


그림 2. NEO IAS의 구조

NEO의 IAS는 스케줄러(scheduler), 커뮤니케이터(communicator), 데이터 풀(data pool), 에이전트(agent)들로 이루어져 있다. IAS는 NEO 시스템 위에 올라간 시스템으로 자율적인 에이전트들로 이루어진다. 다수의 입/출력 모듈을 가지며, 같은 입력정보에 대해서도 상황에 따라 다른 다수의 정보처리 경로를 가진다. 경로에 대한 결과는 상

호정보교환이나 다수 또는 강자 선택 방식의 의사결정 구조를 가지고, 그 경로 중 빠른 처리를 위한 알람(alarm)정보 처리 경로도 가진다. 또한 에이전트들은 다른 에이전트들로부터 오는 정보를 자율적으로 판단하여 처리하고 결과를 다른 모듈에 전달할 수 있다.

에이전트는 목적, 활성화 조건, 내부 변수, 내부 함수, 메인 프로시저로 이루어진다. 내부 함수는 자체 정의 함수, 외부 언어 함수 호출이 가능하고 이를 통하여 규칙기반 모듈, 신경망 모듈, 유전자 모듈, 외부처리 모듈 등을 이용할 수 있다. 각 에이전트들은 직접 에이전트 끼리 정보를 전달하는 것도 가능하지만 데이터 풀에 놓여나진 다양한 정보에 따라 자신의 목적에 부합되는 활성화 조건이 갖추어 지면 정의된 내부변수/함수를 이용하여 메인 프로시저를 구동한다. 에이전트는 스케줄러에 의하여 사이클별로 처리되기 때문에 물리적으로 순차 처리되지만 계산적으로는 병렬 처리된다. 커뮤니케이터는 외부 환경에 대한 시스템 입/출력을 처리함으로써 시스템의 유일성(Identity)을 유지한다. 또한 시스템 내부의 에이전트들도 자체의 입/출력 기능을 가질 수 있다. 데이터 풀은 객체지향형태로 유지되는 공통의 정보 공간이며, 시스템 입/출력 정보의 공간이며, 임의의 조건에서 임의의 에이전트들이 정보를 교환할 수 있는 정보전달 공간의 역할을 한다. 마지막으로 위의 에이전트 정의 방법을 그림 3에 나타내었다.

```
(app-agent is-a agent)
(app-agent rule-system rule-system1)
(rule-system1 is-a rule-system)
(rule-system1 rule-set rule-set1 rule-set2)
(rule-set1 is-a rule-set)
(rule-set rule r1 r2 r3 r3)
(r1 is-a rule)
(r1 if (?x is-a bird))
(r1 then (?x is-a animal)
(app-agent method (:set-height ?value) (
  (printf "previous height => %s"
'((get-list self 'height)))
  (put-list self height ?value)
  (printf "current height => %s"
'((get-list self 'height)))
  )
)
(app-agent1 is-instance-of app-agent)
(app-agent1 height 24)
(app-agent1 :get-height) =>
(app-agent1 :set-height 32) =>
```

그림 3. IAS에서 에이전트 정의 방법

3.3 영상 처리 시스템

본 연구에서는 NEO 시스템에 오픈소스 라이브러리인 OpenCV를 이용하여 간단한 명령어 형태

로 영상처리 할 수 있도록 구축한 시스템을 VisionNEO라 부른다. 스크립트 명령 다이얼로그 박스(그림 4)에서 명령을 수행하며 VisionNEO를 사용하여 자율주행 판단 시스템을 구현한 예를 그림 5에 나타내었으며 그 처리 결과는 그림 6로 나타내었다[6].

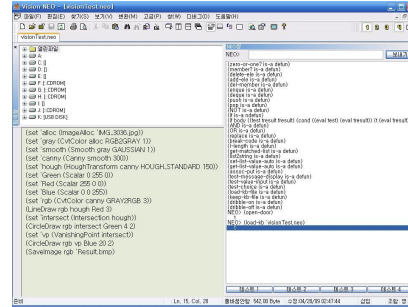


그림 4. 스크립트 명령 다이얼로그 박스 (VisionNEO)

```
(set 'gray (CvtColor img RGB2GRAY 1))
(set 'smooth (Smooth gray GAUSSIAN 1))
(set 'canny (Canny smooth 300))
(set 'hough (HoughTransform canny HOUGH_STANDARD 150))
(set 'rgb (CvtColor canny GRAY2RGB 3))
(LineDraw rgb hough Red 3)
(set 'intersect (Intersection hough))
(CircleDraw rgb intersect Green 4 2)
(set 'vp (VanishingPoint intersect))
(CircleDraw rgb vp Blue 20 2)
(ShowImage rgb 'Result.bmp)
```

그림 5. 비전 처리 시스템 사용 예시

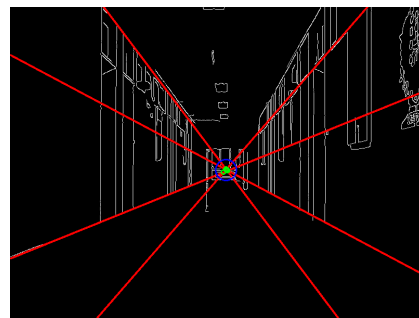


그림 6. 비전 시스템 처리 결과

3.4 지능형 로봇 구성

본 연구에서 VisionNEO의 지능형 로봇의 구성으로 '강아지 로봇'과 '전동 휠체어'를 그림 7에 나타내었다.

강아지 로봇은 무선조종이 가능한 자체의 API를 사용하고, 전동 휠체어는 시리얼 통신을 사용하고 있기 때문에 NEO에서 제어가 가능하도록

명령어를 추가하였다. 강아지 로봇과 전동 휠체어에 장착된 카메라에는 팬틸트(pan-tilt) 기능을 갖추고 있어서 마찬가지로 카메라 제어도 NEO에서 제어가 가능하도록 명령어를 추가하였다.



그림 7. 좌-강아지 로봇, 우-전동 휠체어

IV. 결론 및 향후 연구

본 연구에서는 인공지능 규칙처리 추론엔진을 토대로 한 NEO 시스템에 영상 처리 시스템을 올려 VisionNEO를 개발한다. 그리하여 주변 환경을 이해하고 알맞은 행동을 추론, 학습해 지식을 축적하는 지능형 로봇 비전 프레임워크를 제안한다.

비전 처리는 어려운 접근 방법을 가지고 있었지만 VisionNEO를 이용하여 조금 더 쉽게 접근하도록 하였다. 또한 비전 처리의 결과를 가지고 환경을 조금이나마 이용하여 지능형 로봇이 효율적으로 움직일 수 있는 루틴을 만들어 볼 수 있다. 그렇기 때문에 매뉴얼과 교육교재를 이용하여 교육과 연구가 일체화된 통합 개발 환경으로 거듭날 수 있을 것이다.

인지과학에 기반을 둔 구성주의는 지식이 어떻게 정의되든 사람의 머릿속에 있는 것이며 자신의 경험에 기반을 두고 구성될 수밖에 없는 것이라고 가정한다[7]. 이러한 구성주의적 가설의 배경에는 외적으로 환경으로부터 자연선택과 내적으로 시스템 구성요소들 사이의 자기조직화가 존재한다[8]. 구성주의 입장에서 확장된 지능의 개념은 어떠한 종류의 시스템에도 적용될 수 있다.

구성적 인공지능의 구현에 필요한 중요한 개념 중 하나는 체화된 인지(embodied cognition)이다. 체화된 인지는 인공지능에서 오래 전부터 논의되어 온 그라운드(grounding) 문제이다[9-11]. 그라운드 문제는 구성적 인공지능에서 시스템이 목적을 위하여 주어진 감각 운동기관으로부터 능동적으로 환경을 모델링할 수 있는 기제로, 좀 더 추상적인 모델링에 대한 적절한 토대로서의 역할을 해야 할 것이다.

체화된 인지의 개념을 도입하여 여러 다양한 개념들이 하나의 틀 안에서 정합적인 설명이 어느 정도 가능해지면 구성적 인공지능은 단순히 인간지능에 대한 계산적 이해를 넘어서 인간 삶에 대한 계산적 이해에 도달할 수도 있을 것이며, 이러한 이해만이 인간과 함께 상호작용하는 진정

한 지능적 기계를 구현할 수 있을 것으로 믿는다. 그에 대한 일례로 그림 8에 H-CogAff Architecture를 제시한다[12].

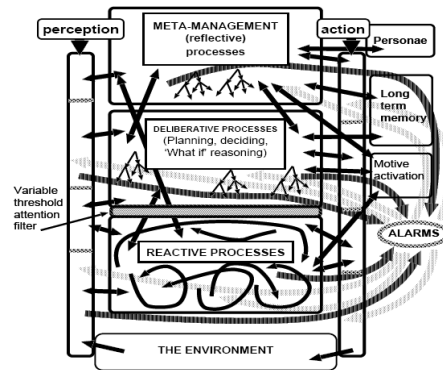


그림 8. H-CogAff Architecture

참고문헌

- [1] 백문홍 외 2명, 정보화시대의 지능형로봇 연구동향, 한국정보과학회 학회지, Vol. 26, No. 01, pp. 46-52, 2008.
- [2] 김영욱 외 2명, 지능 로봇틱스 영상처리 기술동향, 한국정보과학회 학회지, Vol. 26, No. 04, pp. 5-11, 2008.
- [3] 이규만 외 2명, 로봇 소프트웨어 모델링을 위한 관련기술 분석 및 UML과의 호환성에 대한 연구, 한국정보처리학회, Vol. 15, No. 01, pp. 304-307, 2008.
- [4] 이정현 외 2명, 영상처리에서 효율적인 교육과 연구를 위한 비주얼 프로그래밍 개발, 한국정보과학회 논문지, Vol. 32, No. 01, pp. 50-61, 2005.
- [5] NEO 시스템 매뉴얼, 2006
- [6] 장세인 외 8명, 소실점 처리를 이용한 전동 휠체어의 자율 주행 판단, 한국컴퓨터정보학회 동계학술발표논문집, 제 16권, 제 2호, pp. 55-60, 2008.
- [7] 박충식, 구성적 인공지능, 한국 인지과학회 논문지, 제 15권, 제 4호, pp. 61-66, 2004.
- [8] 카우프만, 혼돈의 가장자리, 사이언스북스, 2002.
- [9] Michael L. Anderson, Embodied Cognition: A Field Guide, Artificial Intelligence, 139-159, 2003.
- [10] Tom Ziemke, Rethinking Grounding, Understanding Representation in the Cognitive Science, 1999.
- [11] Rodney Brooks, Intelligence without Representation, Artificial Intelligence, 139-159, 2003.
- [12] Aaron Sloman, Varieties of Affect and the CogAff Architecture Schema, AISB'01, 2001.