



Journal of Knowledge Information Technology and Systems

ISSN 1975-7700

<http://www.kkits.or.kr>

Development of AI Robotic Gomoku Platform Based on Multi-Robot Cooperative System

Sang-Yong Park, Kang-Hee Lee*

Global School of Media, Soongsil University

ABSTRACT

Artificial Intelligence (AI) has become a hot topic since 2016 when Lee Se-dol and AlphaGo became a big hit. However, AlphaGo's computational results were made through the hand of a substitute driver in the game. It predicted that if there was hardware that matched advanced software, viewers would be more immersed in the Go game and that AlphaGo and Lee Se-dol would be clearly visible as AI and human. Therefore, in this paper, we want to design the combination between software and hardware using vision system and Robot for user immersion and real-time interactive communication with AI. In particular, unlike previous experiments in which only one robot was previously treated as a single robot and conducted only in a limited area, it induces higher user interaction through parallel processing in two. The implementation of the vision system took place in the Python environment of Zet Brains, and two robots are used. In fact, when a user starts a concave egg on a board of go, the starting point of the concave AI is calculated through the vision system, and the launch of the board is carried out through the robot. In conclusion, the proposed AI robotic omok platform based on multi-robot cooperative system proves its effectiveness and will be available in areas such as education for children with developmental disabilities.

© 2020 KKITS All rights reserved

KEYWORDS Gomoku, OpenCV, Vision-system, DOBOT, Python, AI, CNN

ARTICLE INFO: Received 21 June 2020, Revised 3 August 2020, Accepted 10 August 2020.

*Corresponding author, Kang-Hee Lee, is with the Global School of Media, Soongsil University, 369 Sando-Ro Dongjak-Gu, Seoul, 06978, KOREA.

E-mail address: kanghee.lee@ssu.ac.kr

1. 서론

2016년 알파고와 이세돌 대국이 ‘지적(知的) 업무’의 인공지능 도입에 대한 인식에 미친 영향 탐색’ 연구와 같이 AI가 일반인들에게도 본격적으로 알려지며 인공지능이란 분야가 각 영역에서 활발하게 연구되고 있다 [1]. 단순히 주어진 업무를 수행하는 1차원적인 로봇이 아니라 상황에 따라 결과값을 연산하는 AI는 마치 인간종과 같이 살아 숨쉬는 개체로 느껴진다.

사실 알파고처럼 연산을 하는 AI는 알파고가 선두주자는 아니다. AI분야는 일반인들의 생각과는 달리 꽤 연구가 오랫동안 진행된 분야이다. 1956년 미국 다트머스 대학에 있던 존 매카시 교수가 개최한 다트머스 회의를 통해 처음으로 인공지능 즉 AI란 용어가 사용되기 시작했으며, 당시의 연구주제는 추론과 탐색을 하는 컴퓨터였다. 인공지능 분야는 1970년대까지 활발히 연구되어, 1969년엔 알버트 린지의 논문을 통해 세계최초로 컴퓨터 바둑 프로그램을 발표하기도 했다 [2].

2000년대까진 DB를 축적한 후 DB를 활용하여 착수점을 찾는 방식이었으나 2006년 이후 몬테카를로 트리 탐색방식으로 비약적 발전을 한다 [3]. 이 알고리즘은 이전까지 있었던 모든 인공지능 바둑 프로그램을 몬테카를로 방식으로 전환하는 선구자적 역할을 수행했다.

이와 달리 알파고는 CNN과 강화학습을 통한 딥러닝 방식이 적용됐다 [4]. 또한 프로기사를 상대로 최초로 승리를 따낸 AI타이틀을 획득했다. 하지만 알파고와 이세돌의 대국에서 직접적인 착수는 알파고가 아닌 대리기사를 통하여 이루어졌다. 소프트웨어의 비약적인 발전이 있었지만 그에 걸맞는 하드웨어적 퍼포먼스는 대국에서 찾아볼 수 없었다. 따라서 본 논문은 AI소프트웨어와 하드웨어를 결합한 로봇 오목 플랫폼을 만들고자 한다.

Python환경에서 비전을 활용하여 유저의 착수점을 인식하고 인식된 결과를 통하여 오목AI는 다음 착수점을 계산한다. 그리고 도봇을 통하여 계산된 착수점에 직접적인 착수를 구현하였다. AI가 그저 연산결과만 도출하는 것 뿐만 아니라 하드웨어적 퍼포먼스로 물리적 착수까지 수행한다면 유저인터랙션 측면에서 더욱 흥미를 이끌 것으로 사료된다 [5].

실험구현을 위하여 컴퓨터 비전은 Python 환경에서 OpenCV 라이브러리를 활용하였으며 도봇의 착수를 위하여 도봇Studio를 사용하였다 [6]. 오목AI는 알파오를 활용하였다 [7].

본 논문은 다음과 같이 구성된다. 제 2장에서는 관련 연구를 소개한다. 관련 연구에서 AI와 하드웨어를 결합한 연구를 소개한다. 제 3장에서는 실험설계를 살펴본다. 이 장에서 비전설계, 비전과 알파오와의 연결 그리고 알파오와 도봇의 연결을 서술한다. 제 4장에서는 3장에서의 설계를 통한 실험구현을 서술한다. 마지막 제5장 결론에선 실험에서 보였던 문제점과 개선방안을 제시한다.

2. 관련 연구

2.1 AI와 도봇을 결합한 기존의 로봇

도봇 Arm은 도봇 Company에서 제작된 하드웨어이다. 2018년 CES Innovation Award를 수상한 이후 로봇팔 관련연구에서 활발하게 사용되고 있다. 도봇 Arm은 EndPoint에 SuctionCup 혹은 그리퍼를 사용하여 물체를 잡고 이동시킬 수 있다. 2장에선 SuctionCup을 활용한 도봇관련 연구를 살펴본다.

- 1) Chinese Chess/Chinese GO3



그림 1 Chinese Chess/Chinese Go3화면
Figure 1. ScreenShot of Chinese Chess/Chinese Go3

Chinese GO3는 도봇을 활용한 바둑게임이다 [8]. 위 실험에선 총1대의 도봇이 활용되었으며 19*19바둑판이 사용되었다. 그리고 위 실험에선 도봇의 End-Point에 Suction-Cup을 장착하여 바둑알을 집는다. 영상에서 확인 할 수 있다시피 도봇의 하드웨어적 길이의 한계로 인하여 제한된 바둑판 영역에서만 대국이 진행된다.

2) OpenCV Tic Tac Toe Engine

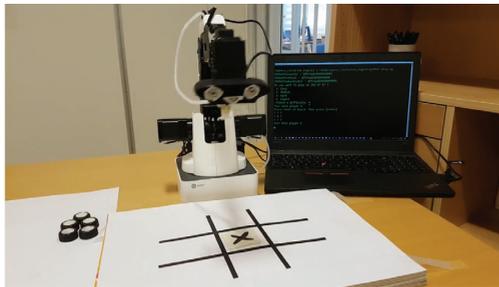


그림 2 OpenCV Tic Tac Toe Engine화면
Figure 2. ScreenShot of OpenCV Tic Tac Toe Engine

Tic-Tac-Toe Engine은 CV환경에서 도봇을 활용하여 Tic-Tac-Toe를 진행한다 [9]. 유저는 착수 후 컴퓨터에 엔터동작을 통하여 비전시스템을 수동적으로 실행시키며 이 후 컴퓨터의 연산결과를 도봇으로 수행하게 된다.

2.2 도봇을 활용한 기존의 실험분석

<그림 1>의 Chinese Go3는 1대의 도봇을 사용하여 실험을 진행하였다. 도봇의 활용가동범위는 도봇의 Center Point로부터 31cm정도인데 이는 19*19 바둑판 끝점과 끝점까지 착수가 불가능하다. 따라서 위에서 살펴본 바와 같이 제한된 영역에서만 대국이 이뤄지게 된다. 이는 도봇 하드웨어의 물리적 한계점이다. 따라서 바둑판 전 영역을 모두 활용하기 위해선 1대의 도봇이 아닌 여러대의 도봇이 활용되어야 하며 이는 협동로봇으로서 병렬형 데이터 처리를 수행해야 한다.

<그림 2>의 Tic-Tac-Toe엔진은 비전시스템을 수동적으로 수행한다. 실시간적 비전시스템이 아닌 이러한 수동적 비전시스템은 사용자로 하여금 게임연속성의 중단을 요구하며 이는 사용자의 몰입도를 낮추게 한다. 따라서 본 논문에서는 2대의 도봇을 활용한 도봇 오목 Platform을 구현하고자 한다. 또한 실시간 비전시스템을 사용하여 사용자가 더욱 게임에 몰입할 수 있도록 유도한다.

플랫폼의 이름은 MRBOT이며 숭실대 Media Robotics Lab을 차용하였다.

3. 다개체 협동 로봇 시스템에 기반한 AI 로봇 오목 플랫폼 MRBOT 설계

3.1 MRBOT Vision 설계

MRBOT의 비전시스템을 개발하기위해 JetBrains의 Python을 사용하였으며 OpenCV라이브러리를 사용하였다. 또한 비전으로 인식할 오목판은 15*15 크기의 오목판을 사용하였으며 1대의 웹캠이 사용되었다.

MRBOT과의 플레이에서 사용자는 백들을 사용

하고 MRBOT은 흑돌을 사용하여 플레이하게 된다.

MRBOT은 사용자의 백돌 착수점을 인식하고 알파오에 인식좌표를 전송한다. 이후 알파오에서의 연산값을 도봇에 전송하여 최종적으로 도봇을 통하여 최종적인 착수가 이루어진다. 따라서 비전시스템에선 백돌만 고려된다는 점을 밝힌다.

흰색영역검출을 위하여 흰색의 최댓값,최소값 그리고 검은색의 최댓값과 최소값을 numpy배열로 정의한다.

두 영역의 RGB값을 HSV로 변환하여 마스킹한 후 비전영역에 들어오는 흰색영역의 범위만 검출하도록 시스템을 설계하였다.

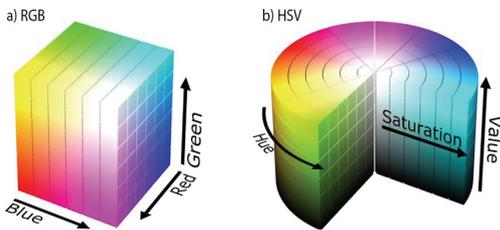


그림 3 RGB영역과 HSV영역
Figure 3. RGB scale and HSV scale

그리고 검출된 영역의 위치정보를 알파오에서 읽을 수 있도록 TextFile에 좌표포지션을 저장한다.

3.2 MRBOT 오목프로그램 설계

MRBOT의 오목프로그램은 1장에서 서술한 바와 같이 알파오를 활용하였다. 알파오는 Python환경에서 작성되었다. 알파오의 데이터 학습방식은 CNN 방식을 사용한다. 유저와 유저 간의 대국 정보를 PNG파일로 저장하고 이를 학습데이터로써 활용한다.

학습 데이터셋은 9개의 히든레이어를 통과한다.

이 때 설정되는 가중치는 1개이며 9개의 히든레이어에서 유의미한 데이터를 수집하기 위하여 렐루함수를 사용한다. 렐루함수는 0 미만의 값은 패

스하고 0 이상의 값을 출력하여 시그모이드 함수보다 유의미한 데이터를 갖는다. <그림 4>는 렐루함수의 그래프이다 [10].

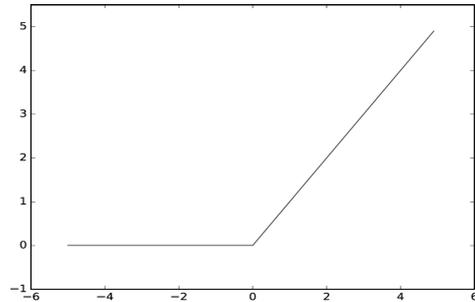


그림 4 렐루함수 그래프
Figure 4. graph of Relu Function

렐루함수로 학습시 100에폭까지 머신러닝을 수행하여 AI가 착수할 상황별 최적의 위치를 데이터셋 기보를 통하여 학습시킨다.

AI와 사용자와의 대국에선 비전부분에서 인식한 위치정보를 TextFile에 저장하면 알파오는 텍스트파일을 열고 위치 정보를 읽는다.

읽은 위치 정보에 대응하는 점으로 마우스를 이동시켜 마우스 클릭 이벤트로 사용자 착수점에 백돌을 착수하게 되며 이에 대응하는 AI의 착수점을 연산하여 게임상 흑돌 착수를 수행한다. 그리고 착수포지션을 도봇 파트에서 읽을 수 있도록 TextFile로 저장한다.

3.3 MRBOT 도봇 설계

도봇은 15*15오목판 전 영역을 활용하기 위하여 2대의 도봇이 사용되었다. 그리고 1번 은 좌표점 (1,1)부터 (8,15)까지 영역을 담당하고 2번 은 (9,1)부터 (15,15)까지의 영역을 담당한다.

<그림 5>는 MRBOT의 동작흐름도이다. 알파오에서 전달받은 위치에 대응하는 좌표점으로 흑돌을

집어서 착수하게 된다. 이는 <그림 5>의 마지막단계에 해당한다.

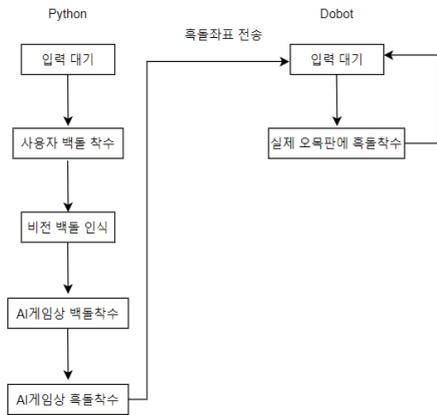


그림 5 MRBOT의 동작흐름도
Figure 5. Flow Chart of MRBOT.

4. MRBOT 구현 및 실험

4.1 MRBOT Vision과 오목게임 구현 및 실험

이브러리를 활용하였다. 비전시스템은 실시간으로 작동하며 사용자의 백돌만 인식하되 도봇이 착수하는 흑돌은 인식하지 않도록 하였다.

<그림 6>은 흑돌 착수 시 웹캠화면과 CV화면이다. 웹캠 상에 흑돌을 착수 한 걸 확인할 수 있으며 비전 시스템상에선 흑돌 착수시 Detect 처리를 하지 않았기 때문에 흑돌 영역이 검출되지 않는 걸 확인할 수 있다.

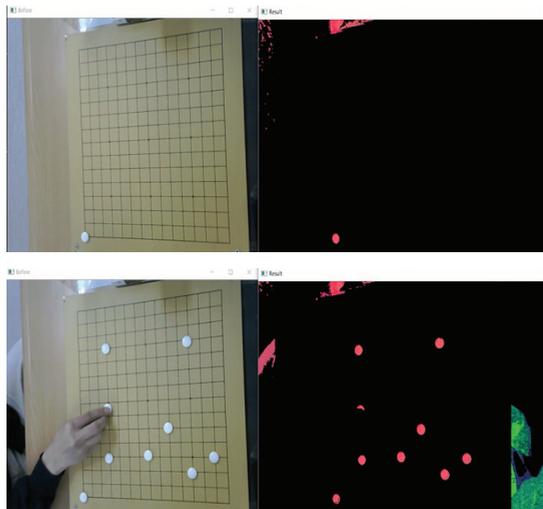


그림 7 백돌착수시 CV영역검출
Figure 7. Image of When Detect White Stone.

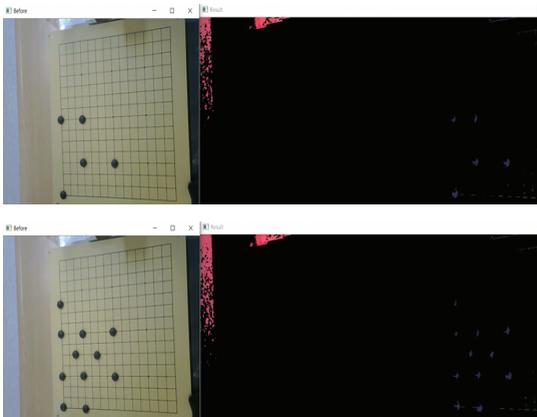


그림 6 흑돌착수시 CV영역검출
Figure 6. Image of When Detect Black Stone.

vision시스템에서 백돌인식을 위하여 openCV라

<그림 7>은 사용자의 백돌 착수 시 영역검출 화면이다. 실시간으로 유저의 백돌을 인식하며 색인식은 마스킹한 흰색범위 값이 들어올 때 수행된다.

본 논문에선 흰색 범위 값을 $[0,0,195] \sim [0,0,0]$ 으로 설정하였다. 이렇게 인식한 위치좌표를 TextFile에 값을 저장하여 알파오 프로그램에서 읽을 수 있도록 설계하였다.

<그림 8>과 같이 알파오는 TextFile에 저장된 백돌좌표를 읽고 오목 게임판에 그에 대응하는 점을 마우스로 클릭하여 착수하게 된다. 이는 Python에서 제공하는 'pyautogui를 import하여 moveTo와 Click이벤트로 구현하였다 [11].

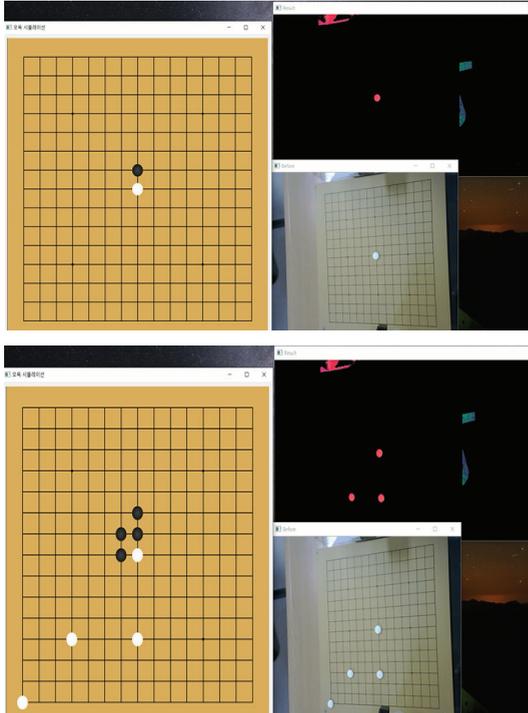


그림 8 비전프로그램과 알파오 연결
Figure 8. Connect vision with AlphaO

반대로 마우스이동을 통하여 도봇제어도 가능하다. 마우스와 눈동자의 움직임을 연결하여 게임상 오목판에 착수시 이를 도봇으로 실제 오목판에 착수 또한 가능할 것으로 보이며 이는 전신 마비 장애인들의 오목수행으로 확대까지 가능한 점을 시사한다 [12].

4.2 MRBOT 도봇 구현

도봇은 도봇Script에서 작성되었다. 도봇을 움직이기 위해서 예비동작들이 들어가게 된다. 이는 Via-Point이며 Via-Point를 거치지 않게 설정하면 도봇의 MoveError가 발생하는 것을 확인하였다. 따라서 바둑알을 suctionCup으로 잡고 착수까지 총 6개의 via-Point를 설정하였다.

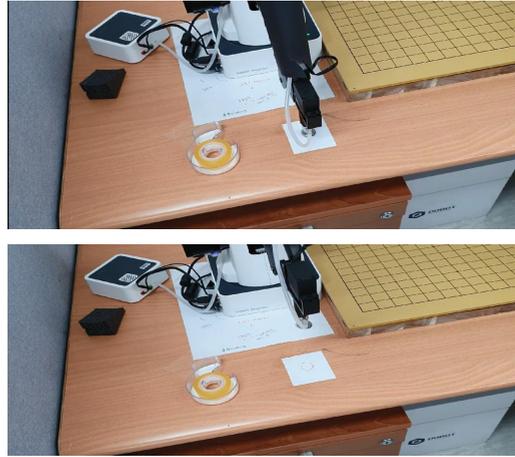


그림 9 도봇의 바둑알 집기
Figure 9. Pick a stone by Using Dobot

<그림 9>와 같이 도봇은 검정바둑알을 집고 알파오로부터 입력받은 좌표로 바둑알을 이동하여 착수한다. 바둑알을 집는건 SuctionCup으로 구현하였다.

위 그림은 (1,1)부터 (8,15)까지의 위치를 담당하는 1번 도봇이다. 각 좌표는 위치좌표와 1대1맵핑을 시켜 도봇의 착수정확도를 높였다. 또한 2대의 도봇을 사용함으로써 오목판 전 영역에 오목알을 착수 할 수 있게 설계하였다.



그림 10 (1,2)점으로 착수
Figure 10. Set a Stone to (1,2)Position

<그림 10>은 (1,2)까지의 착수를 파트로 나누어 캡처한 사진이다. 바둑알을 집고 착수하기까지 6개의 via-point를 거친다. via-point는 함수로 구현하여 오목판의 각 위치좌표만 파라미터로 입력받도록 구현하였다.

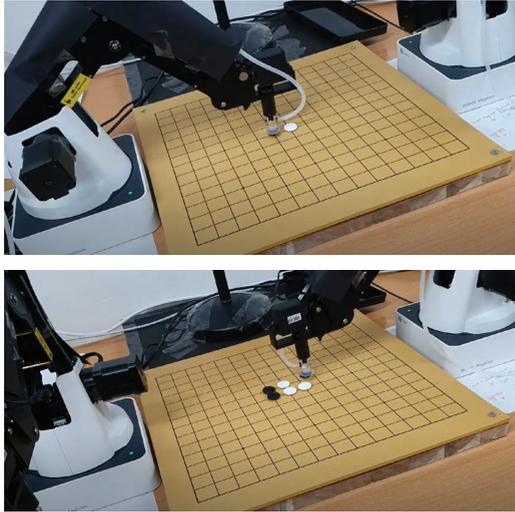


그림 11 2대의 도봇동작모습
Figure 11. Picture of Operating 2 도봇s

<그림 11>은 최종적으로 2대의 도봇이 작동하는 모습이다. 1번 도봇과 2번 도봇은 실시간 While문으로 반복문을 돌리며 알파오로부터 위치좌표를 연속적으로 입력받는다. 따라서 별도의 처리를 하지 않으면 동일한 입력값을 끊임없이 수행한다. 따라서 본 논문에서는 한번 위치좌표를 입력받으면 동일한 위치좌표로 다시 동작을 수행하지 않도록 구현하였다. 또한 게임의 종료를 위하여 overCheck란 텍스트파일을 만들어 게임이 끝나면 문자열 1을 전달받아 모든 루프문이 종료되도록 설계 구현하였다.

4.3 실험 최종 구현

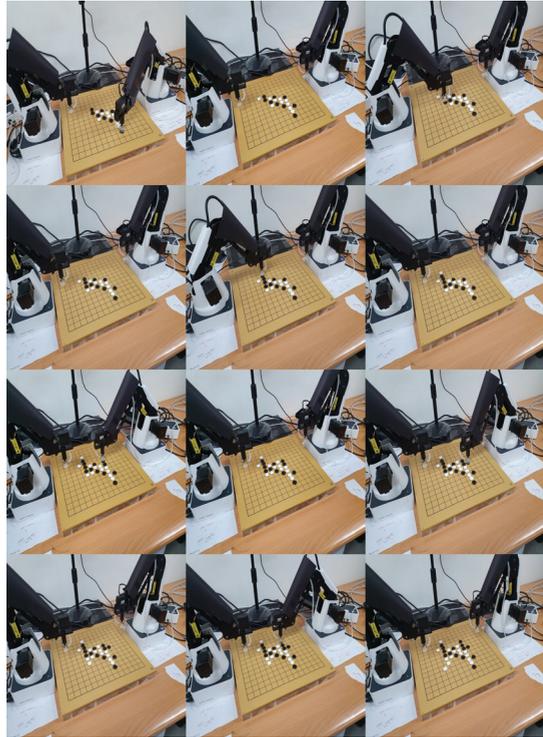


그림 12 MRBOT 최종구현
Figure 12. Final Implementation of MRBOT

<그림 12>는 오목게임을 끝까지 진행한 사진이다.도봇 2대는 어려없이 오목게임을 수행하였으며 오목게임 종료 시 overCheck 텍스트파일을 확인하여 루프문을 종료하는 것 까지 확인하였다.

5.결 론

실험에서 보인 문제점은 다음과 같다. 본 논문의 Vision 부분에선 비전시스템을 실제 오목판과 1대1 맵핑을 시켰다. 하지만 웹캠이 각도가 초기값과 달라질 경우에 비전시스템의 오차가 발생하는 것을 파악하였다. 따라서 이를 보완하기 위하여 맵핑의 Step-Size를 10Pixel로 설정하였다. 하지만 이를 더 보완하기 위한 방법은 후술한다.

도봇은 미리 세팅된 오목알을 집고 좌표에 착수

한다. 따라서 이 후 오목알을 도봇이 착수했다면 새로운 오목알을 다시 위치시켜줘야 한다. 따라서 플레이어는 오목도 착수하여야 하고 도봇의 오목알 까지 착수해야 하는 번거로움이 발생한다. 이는 사용자의 몰입도를 크게 저하시킨다.

앞서 본 논문에서 설계 구현했던 Vision 시스템은 설정해놓은 세팅값을 벗어나면 인식에 오차가 생기는 문제점이 있었다. 이는 통합된 플랫폼환경 구축에 있어서 큰 문제점이다.

따라서 어느 각도에서든 인식하기 위하여 앞서 구현한 비전 프로그램처럼 오목알의 위치좌표를 1대1맵핑으로 검출하는게 아닌 바둑판의 최외각 네 점을 검출하여 좌표를 맵핑시키고자 한다. 이는 추후에 개선된 소프트웨어모델에서 구현할 예정이다 [13].

또한 도봇이 사용할 오목알을 사용자가 직접 놓아 주어야 했던 기존의 방식이 아니라 오목알을 공급 받을 수 있는 모듈을 도봇에 장착하는 방식으로 개선하려 한다.

결과적으로 개선된 비전시스템을 통하여 오목뿐만 아니라 바둑AI를 이용한 게임플랫폼 개발도 가능할 것으로 보인다 [14].

References

- [1] D. W. Kim, and B.J. Kim, *How alphago does change peoples perception of introduction of Artificial Intelligence into Intellectual work?*, Journal of Cybercommunication Academic Society, Vol. 33, No. 4, pp. 107-158, 2016.
- [2] A. L. Zobrist, *A model of visual organization for the game of go*, AFIPS '69(Spring):Proceedings of the May 14-16, 1969, spring joint computer conference, 103-112, 1969.
- [3] S. G. W. Munos, and O. Teytaud, *Modification of uct with patterns in monte-carlo go*, Technical Report RR-6062, Vol. 32, pp. 30-56, 2006.
- [4] D. Silver, A. Huang, A. Huang, and C. J. Maddison, *Mastering the game of go with deep neaural networks and tree search*, Nature 529, pp. 484-489, 2016.
- [5] K. G. Bhan, and H. H. Kim, *A study on the effects of experience based physical interface on game flow*, Archives of Design Research, Vol. 21, No. 3, pp. 201-210, 2008.
- [6] Computer-Vision and Open CV, <https://github.com/opencv/opencv>, May 2020.
- [7] d a i l y - c o d i n g , <https://sites.google.com/view/dailycoding00/main/alphaoh-learning>, May 2020.
- [8] Dobot, https://www.youtube.com/watch?v=-ZIzyfP18_Q, May 2020.
- [9] Vegard Haugland, https://github.com/v3gard/opencv_tictactoe_engine, May 2020.
- [10] K. He, X. Zhang, J. Sun, and S. Ren, *Delving deep into rectifiers: surpassing human-Level performance on imageNet classification*, ICCV '15: Proceedings of the 2015 IEEE international Conference on Computer Vision(ICCV), pp. 1026-1034, 2015.
- [11] Pyautogui, <https://pyautogui.readthedocs.io/en/latest>, May 2020.
- [12] G. W. Kim, M. S. Yang, and J. Y. Yang *Construction of robot control environment using pupil mouse*, Korea Knowledge Information Technology Society, Vol. 8, No.

6, pp. 1-8, 2013.

- [13] Y. K. Yoon, and Y. G. Lee, *Algorithm of detecting go board corners*, The Korean Society Of Broad Engineers, pp. 223-236, 2017.
- [14] S. O. An, S. K. Kim, R Hwang, N. S. Jang, B. Han, and J. H. Kang, *Efficiency 3D baduk game using directX and Ai*, Korea Knowledge Information Technology Society, Vol. 4, No. 2, pp. 45-52, 2009.

다개체 협력 시스템 기반 AI 로봇 오목 플랫폼 개발

박상용¹, 이강희²

¹승실대학교 글로벌미디어학부 학사과정

²승실대학교 글로벌미디어학부 부교수

요 약

2016년 이세돌과 알파고의 대국 이후로 AI는 큰 화두가 되었다. 하지만 대국에서 알파고의 연산결과는 대리기사의 손을 통하여 이루어졌다. 발전된 소프트웨어에 걸맞는 하드웨어가 있었다면 시청자로 하여금 더욱 바둑대국에 몰입하게 되고 알파고와 이세돌의 대국이 AI와 인간의 대국으로 극명하게 가시화 될 것이라 예측했다. 따라서 본 논문에서는 사용자 몰입도, AI와의 실시간 쌍방향 소통을 위하여 비전 시스템과 도봇을 사용하여 소프트웨어와 하드웨어간의 결합을 설계구현 하고자 한다. 특히 기존에 1대의 도봇으로 처리되어 국한적 영역에서만 진행되었던 이전 실험과 달리 2대의 도봇 환경에서 병렬적 처리를 통하여 보다 높은 사용자 인터랙션을 유도한다. 비전 시스템의 구현은 Zet Brains사의 python 환경에서 이루어졌으며 2대의 도봇이 사용된다. 실제 바둑판에서 유저가 오목알을 착수할 시 비전 시스템을 통하여 오목AI의 착수점이 계산되고 이는 도봇을 통하여 바둑판에 착수가 이루어진다. 결론적으로 본 논문에서 제안된 다개체 협력 시스템 기반 AI 로봇 오목 플랫폼을 활용하여 발달 장애 아동 교육과 같은 치료 및 교육 분야에서 활용할 수 있을 것이다.

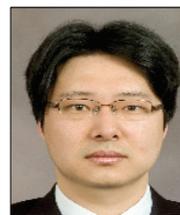
감사의 글

이 논문은 2017년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초 연구사업임. (NRF-2017R1D1A1B05036028). 이 논문은 2019년 대한민국 교육부와 한국연구재단의 인문사회분야 중견연구자지원사업의 지원을 받아 수행된 연구임 (NRF-2019S1A5A2A01043252)



Sang-Yong Park is in the bachelor's degrees in Global School of Media from Soongsil Univeristy, Korea. He belongs to the Media-Robotics Laboratory, Soongsil University. He is interested in media robotics and SAP.

E-mail address: qwer5766@naver.com



Kang-Hee Lee received the B.S., M.S. and Ph.D. degrees in EECS from KAIST, Korea, in 1999, 2001 and 2006, respectively. Dr Lee is currently an Associate Professor in the Department of Digital Media, Soongsil University. He is interested in robotic intelligence & emotion and media robotics.

E-mail address: kanghee.lee@ssu.ac.kr