

WESC-17-10-결과요약서

지능형 휴머노이드 부문 결과요약서

참가부문: 지능형 휴머노이드

팀 명: KOBOT-R

팀구성원

No.	구분	성명	소속명	No.	구분	성명	소속명
1	팀장	김예린	국민대학교 컴퓨터공학부	6	팀원		
2	팀원	차민준	국민대학교 컴퓨터공학부	7	팀원		
3	팀원	구민준	국민대학교 컴퓨터공학부	8	팀원		
4	팀원	박형준	국민대학교 컴퓨터공학부	9	팀원		
5	팀원	송영은	국민대학교 컴퓨터공학부	10	팀원		

2017. 10. 10.

1. Software 구조

1.1. 하키 경기

- 모터 제어 모듈: 모터 제어 보드로 명령을 Write하는 모듈
- 센서/통신 모듈: 로봇 간 통신을 위해 동기화하는 모듈
- 패스 모듈: 통신을 통해 Puck의 패스를 수행 하는 모듈
- 슈팅 모듈: 영상을 처리하여 패스 받은 Puck을 컨트롤 및 슈팅하는 모듈

1.2. 컬링 경기

- 카메라 관련 모듈: 카메라 버퍼를 처리하는 모듈
- 모터 제어 모듈: 모터 컨트롤러에 명령을 전달하고 확인하는 모듈
- 이동 모듈: 스톤의 위치를 확인하여 로봇을 스톤까지 이동시키는 모듈
- 비교 모듈: 스톤과 표적을 비교하여 로봇의 위치를 재조정하는 모듈
- 슈팅 모듈: 영상을 처리하여 스톤을 슈팅하는 모듈
- AI 모듈: 전체 모듈과 프로그램의 제어를 담당하는 모듈

2. 구현 상의 제약 조건들과 극복 방안

2.1. 반사광 문제

문제점: YCbCr 4:2:2 색 포맷은 조도에 민감하여 광량이 변화하면 색을 인식하기 어려움

해결방안:

- 조도에 덜 민감한 HSV 색 포맷을 사용
- 미션 수행에 필요한 Puck과 스톤의 모양이 모두 원형이라는 점에 착안하여, Opencv 라이브러리 활용하여 회색조로 변환한 영상으로부터 원모양을 인식 및 추출하여 활용

2.2. 로봇의 위치 파악 문제

문제점: 미션 경기 수행 중, 로봇이 자신의 위치를 파악할 수 없는 어려움

해결방안:

- 위치가 고정된 대상(표적, 골대)과 로봇의 움직임을 바탕으로 위치 추론

3. 쓰레드 또는 프로세스들 간의 통신 방법

3.1. 하키 경기

- 첫 번째 로봇이 패스를 수행하면 슈팅을 수행할 로봇에게 메시지를 전달
- 대기 중인 로봇의 영상에 Puck이 감지되면 슈팅 모듈로 이행

3.2. 컬링 경기

- 로봇간의 통신은 불필요하며, 로봇 내부 모듈 간에만 충돌이 없도록 설계한다.

4. 사용된 알고리즘에 대한 설명

4.1. 하키 경기

4.1.1. Hough transform

하키 경기에서 로봇이 Puck의 위치를 확인하기 위한 알고리즘. 허프 변환을 이용하여 원을 검출하고 이를 기반으로 Puck의 위치를 판단 및 각각의 모듈을 수행한다.

4.1.2. 라인 탐색 알고리즘

경기장의 특성에 착안하여 라인을 검출하고 라인 안쪽에 ROI를 설정하여 탐색 시간을 최소화하였다.

4.2. 컬링 경기

4.2.1. Hough transform

컬링 경기에서 로봇이 스톤의 위치를 확인하기 위한 알고리즘. 허프 변환을 이용하여 원을 검출하고 이를 기반으로 스톤의 위치를 판단 및 각각의 모듈을 수행한다.

4.2.2. 거리 측정 알고리즘

x, y 좌표가 주어졌을 때, 로봇과 좌표상의 대략적인 물리적 거리를 계산한다.

4.2.3. 위치 비교 알고리즘

스톤과 표적간의 x, y 좌표를 이용하여 기울기를 구하고 이를 기반으로 로봇과 스톤, 표적이 일직 선상에 놓이도록 하여 스톤의 슈팅을 수행한다.

5. 개발 방법

5.1. 개발 환경

개발 언어: C언어, 로보베이직

운영체제:

- Raspbian(C 언어)
- Windows10 (로보베이직)

5.2. 알고리즘 개발을 위한 테스트 환경 구성

G-Editor와 Opencv를 활용하여 이미지 데이터를 처리하는 알고리즘들의 결과를 가시적으로 볼 수 있도록 구성하였다.

6. 시험 방법

6.1. 하키 경기

- 1차 시뮬레이션: 공통 - Puck을 인식하도록 한다.
- 2차 시뮬레이션: 첫 번째 로봇 - Puck과 로봇의 위치를 인식하도록 한다.
- 3차 시뮬레이션: 첫 번째 로봇 - 패스를 수행한다.
- 4차 시뮬레이션: 두 번째 로봇 - Puck과 골대의 위치를 인식하도록 한다.
- 5차 시뮬레이션: 두 번째 로봇 - 패스를 받은 Puck으로 슈팅을 수행한다.
- 6차 시뮬레이션: 일련의 과정을 연결한다.
- 7차 시뮬레이션: 기록을 단축한다.

6.2. 컬링 경기

- 1차 시뮬레이션: Puck을 인식하도록 한다.
- 2차 시뮬레이션: 표적을 인식하도록 한다.
- 3차 시뮬레이션: Puck을 슈팅하기 위한 알맞은 위치까지 로봇을 이동시킨다.
- 4차 시뮬레이션: Puck과 표적의 위치를 비교하여 슈팅하도록 한다.
- 5차 시뮬레이션: 일련의 과정을 연결한다.
- 6차 시뮬레이션: 기록을 단축한다.

7. 기술적 차별성

7.1. Thread 역할 분리

프로그램의 주요 실행흐름에서 장치들과 소통하는 루틴을 Thread로 분리하였다.

- 1) I/O 지연 최소화
- 2) 빠른 비디오 버퍼 갱신
- 3) 실시간 원 검출

7.2. 단계별 회전 정도 세분화를 통한 성능 개선

영상 데이터에서 Puck과 스톤, 목표물의 위치 좌표에 따른 기울기에 따라 수행할 동작을 세분화함으로써 명중률을 높이고 수행 시간을 최소화하였다.

7.3. 영상 처리 성능 최적화

7.3.1. 부분탐색을 통한 성능 개선

화면 모두를 탐색하는 것이 아닌 가장 있을 가능성이 높은 위치부터 나누어 탐색하며 원을 발견했을 경우 나머지 부분을 탐색하지 않는 방법으로 성능을 향상시켰다.

7.3.2. 라인탐색을 통한 성능 개선

경기장의 특성에 착안하여 영상 데이터에서 라인을 검출하여 라인 안쪽에 관심영역을 두고 목표물을 검출하도록 하였다.