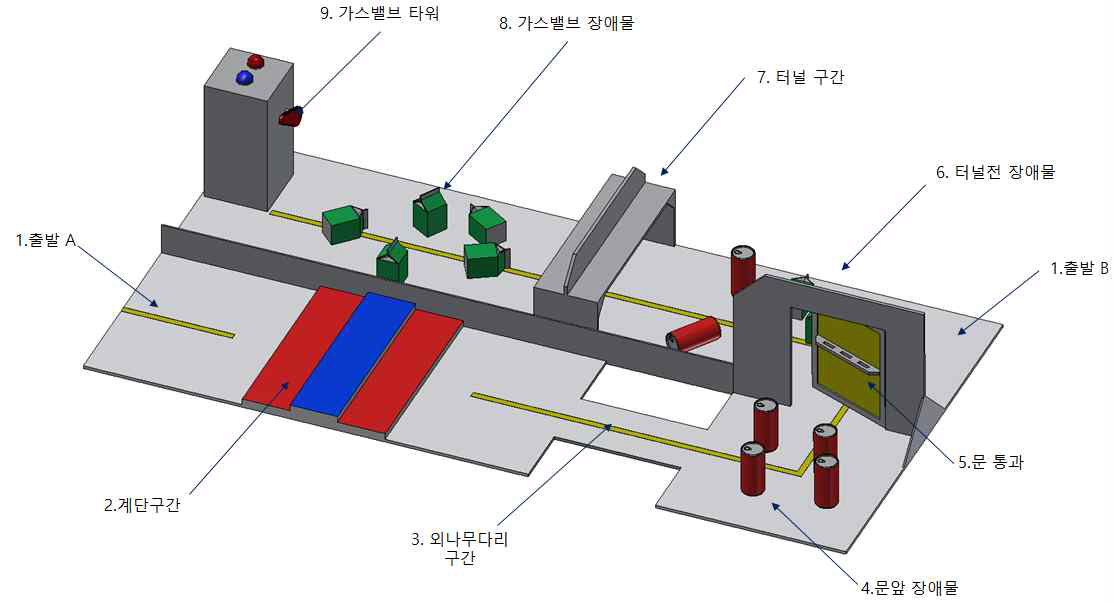
1. **개요** 
   1. **작품 개요**:

휴머노이드(humanoid : 인간형 로봇)란, 인간의 형태를 모습으로 한 로봇을 의미한다. 형태뿐 아니라, 인간과 같은 인식기능, 운동기능을 구현하기 위해서는, 로봇기술의 총체적 발전이 궁극을 이루어야 하기 때문에 가장 고난도의 지능형 로봇이라 할 수 있다. 우리는 이번 대회에서 주어진 휴머노이드를 조종하기 위한 소프트웨어를 제작하고 이를 통해 모조재난현장모형에서 휴머노이드를 조종하여 미션을 수행 할 것이다.

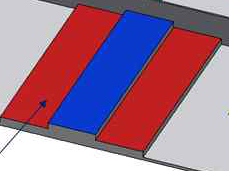
재난 구조 경기 방식

경기 개요

2대의 2족 보행 로봇을 각각 움직일 수 있게 하는 제어 소프트웨어 및 통신 기술과 관련된 소프트웨어를 개발하고, 각 로봇을 갖고 각 회당 2번의 주어진 미션을 안정적으로 수행하여야 한다. 우리 팀은 총 장애물 코스를 5단계로 구분하여 정리했다.



* 제 1코스: 빨파빨 계단

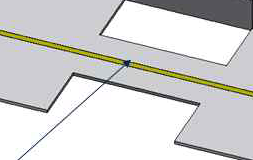


색상: 빨간색, 파란색

규격: 약 600mm(가로) \* 150mm(세로) \* 10mm(높이)

목표: 로봇이 계단을 인식하고 넘어지지 않고 진행하도록 소프트웨어 및 통신기술을 이용하여 장애물을 통과해야 한다.

* 제 2코스: 외나무 다리

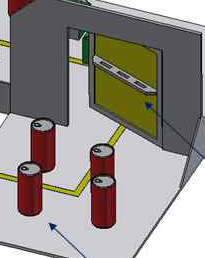


가이드라인 색상: 노란색 라인 두께 약 20mm

규격: 약 400mm(길이) \* 200mm(폭)

목표: 좁은 다리를 로봇이 떨어지지 않게 프로그래밍하여 장애물을 통과해야 한다.

* 제 3코스: 문

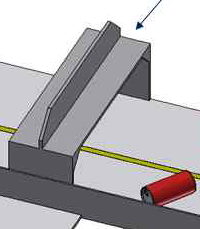


문 색상: 노란색 혹은 초록색

규격: 문 높이 390mm \* 너비 300mm

목표: 로봇이 닫혀있는 문을 열고 다음 단계로 진행할 수 있도록 프로그래밍 해서 통과해야 한다.

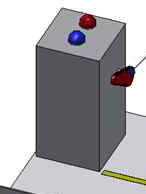
* 제 4코스: 터널



규격: 터널 높이 200mm \* 너비 300mm

목표: 로봇의 높이보다 낮은 터널을 로봇이 앉거나 혹은 엎드려서 진행할 수 있도록 프로그래밍하여 장애물을 통과해야 한다.

* 제 5코스: 가스밸브 타워



규격: 구조물 높이 260mm \* 너비 160mm (밸브 레버 높이 \* 래버 길이)

목표: 전 장애물을 모두 통과하고 난 뒤 로봇이 밸브를 내리도록 한다. 밸브를 내리면 미션이 완료된다.

* 1. **필요성**

재난 현장에 로봇을 투입하는 데에는 크게 두 가지 목적이 있다. 첫째로 부족한 인력을 보충하기 위해서다. 예를 들어 약 6,300명의 사망자가 발생한 고베 대지진과 같은 대규모 재난의 경우에, 사고 현장에 동원할 수 있는 소방관과 경찰관의 숫자가 부족했고, 이것을 계기로 일본 재난 로봇의 대가 타도코로 사토시(Sathosh iTadokoro) 교수는 로봇 개발에 나섰다고 말했다. 그러나 대형 재난의 경우 일어나는 빈도 수가 많지 않기 때문에 이는 특별한 경우에 해당하고 다음 두 번째 이유가 주 이유이다. 

(다음 사진은 성화봉송에 참여한 휴보가 가상재난환경에서 벽을 뚫는 이벤트를 진행하고 있는 사진이다. )

두 번째는 인간이 직접 들어가기 어렵거나 위험한 사고 현장에 인간 대신 로봇을 투입하기 위한 것이다. 아직 건물 붕괴 위험이 남아 있거나 잔여 화학 물질 위험에 노출되어 있을 경우에, 인간을 구조요원으로 직접 투입한다면 그 구조요원의 생명에 치명적인 위협이 될 수 있기 때문이다. 실제로 2011년 동일본 대지진으로 인해 후쿠시마 제1 원자력 발전소에 원전 전원상실 사고가 일어나 방사능이 유출되었을 때, 무인기가 원자로 건물 상부를 정찰했고 인간을 대신하여 로봇이 건물에 들어가 내부를 조사했다. 그 결과 적은 인명피해로 내부조사를 마칠 수 있었다. 만약 실제로 사람이 들어가서 직접 조사했다면 적지 않은 후유증을 남겼을 것이다. http://linkback.hani.co.kr/images/onebyone.gif?action_id=217238594487a7194f862757a3a947a이처럼 인원이 부족하거나 사람이 하기 힘든 일을 휴머노이드를 사용하게 된다면 비교적 효과적으로 상황을 대처 할 수 있기 때문에 재난 상황에서 휴머노이드 사용해야 한다.

* 1. **개발목표**

이번 프로젝트의 목표는 가상재난 현장에서 휴머노이드가 미션을 모두 수행하고 최대한 안정적이고 빠른 시간 내에 가스밸브타워의 밸브를 내리도록 하게 하는 것이다.

세부목표

- 로봇의 2족 보행을 이용한 임베디드소프트웨어 개발

- 제공되는 로봇이 미션을 잘 수행할 수 있도록 해주는 적절한 알고리즘과 소프트웨어 개발

- 영상처리 기술 및 로봇제어를 위한 소프트웨어 개발

- 제공된 로봇이 각기 다른 움직임을 잘 소화할 수 있도록 하기 위한 제어 소프트웨어 및 통신기술과 관련된 소프트웨어 개발

**2. 개발방법**

**2.1. 기술적 요구사항**

**∙ 미션 수행 구동체(휴모노이드) 사양: MF-RAPI3**

|  |  |  |
| --- | --- | --- |
| **<MF-RAPI3>** | 크기 및 무게 | 100mm x 190mm x 410mm, 1.4 kg |
| 관절 수 | 18 관절 |
| 배터리 | NIMH 6V 1500mA |
| 카메라 | 5M픽셀(1080P 동영상 촬영가능) |
| 동작시간 | 약 20분 내외 |
| 2축 기울기 센서 | 넘어짐 감지 |
| 2축 자이로 센서 | 자세 및 동작 안정성 확보 |
| 두뇌보드 및 두뇌보드 OS | Raspberry Pi 3 Model B, 라즈비안(Respbian) 리눅스 |
| 제어보드 및  제어보드 OS | MR-C3024FX,  로보베이직 |

**∙ 미션 수행 임베디드 보드: 라즈베리파이**

|  |
| --- |
|  |
| **<Raspberry Pi 3 Model B>** |
| 1,2Ghz 쿼드코어, 1GB RAM + (16GB 메모리 탑재, 무선 랜 및 Bluetooth 포함) |

**∙ 미션 수행 소프트웨어: 로보베이직**

|  |
| --- |
|  |
| **<로보베이직 v2.8>** |
| 베이직 문법 기반 language, 모션 캡쳐를 이용한 모션작성 및 알고리즘 작성에 적합 |

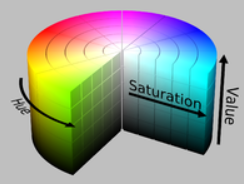
**∙ 영상처리**

1. OpenCV와 python



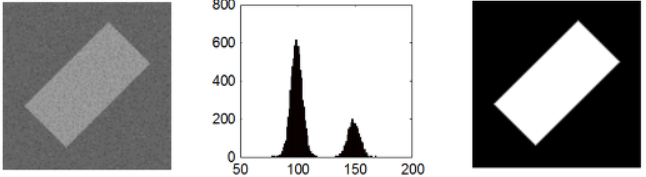
: OpenCV는 실시간 이미지 프로세싱 및 [컴퓨터 비전](https://ko.wikipedia.org/wiki/%EC%BB%B4%ED%93%A8%ED%84%B0_%EB%B9%84%EC%A0%84)을 목적으로 한 프로그램밍 [라이브러리](https://ko.wikipedia.org/wiki/%EB%9D%BC%EC%9D%B4%EB%B8%8C%EB%9F%AC%EB%A6%AC)이다. 이와 더불어 동적 타이핑 대화형이자 인터프리터식 언어인 파이썬을 이용하여 알고리즘을 프로그래밍한다.

1. 색상모델-HSV



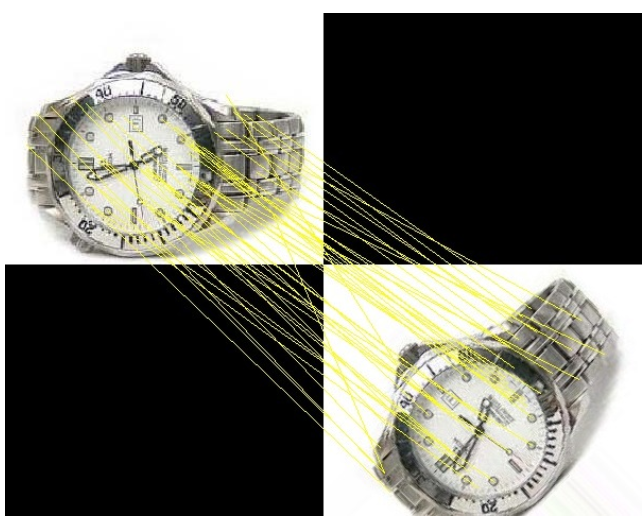
: HSV 모델은 RGB 모델이 갖는 한계점을 보완하기 위해 사용하며 H는 색상값으로 0°~360°의 범위를 갖는 가시광선 스펙트럼의 원기둥형 배치를 갖는다. S 는 채도값으로 0~100%의 범위를 갖으며 0%가 무채색을 나타낸다. V는 명도값으로 0~100%의 범위를 갖고 검은색이 0%이다.

1. 전역고정이진화



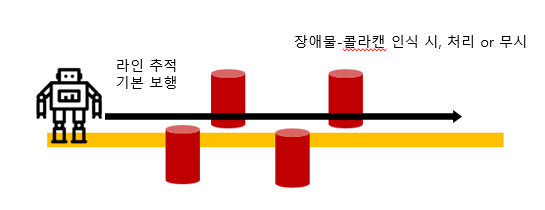
: 영상처리에서 이진화는 어떤 주어진 임계값(threshold)보다 밝은 픽셀들은 모두 흰색으로, 그렇지 않은 픽셀들은 모두 검은색으로 바꾸는 것을 지칭한다. 아래 예는 이진화 문제 중 가장 단순한 형태로서 균일한 밝기를 갖는 배경과 물체에 약간의 노이즈를 섞어서 생성한 영상이다.

1. 물체 인식-Surf



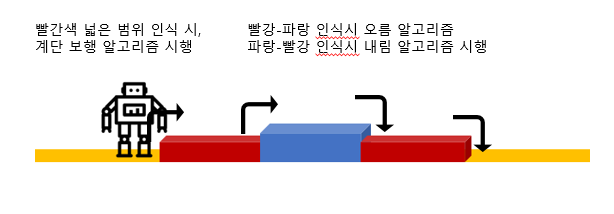
: Surf 알고리즘은 Speed-Up Robust Feature의 약어로 SIFT 알고리즘보다 속도를 향상시키고 특징을 더 잘 추출 가능하다. 특징점 추출, 주 방향 결정, descriptor 생성으로 진행되며 박스필터와 적분영상을 사용한다.

* 1. **개발방법**
* 진행 방식: 라인트레이서 및 방해물 처리



: 기본 보행 알고리즘은 기존 소스에서 가져온다. 라인을 인식 하지 못하는 경우 멈춰서 라인 추적 알고리즘을 시행한다. 장애물-콜라캔 인식은 빨간색 범위의 비율로 판단한다. 실험을 통해 장애물을 치울지 무시하고 주행을 실시할지 정한다.

* 제 1코스



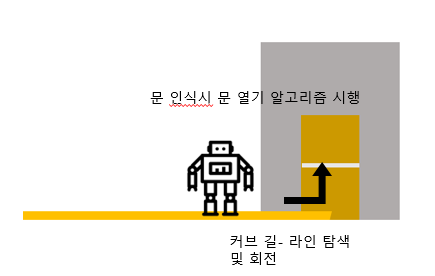
: 빨간색 범위의 비율로 빨간색 계단을 인지한다. 계단 보행 알고리즘은 기존 소스를 참고하고 모션 캡쳐로 맞춘다. 오름 알고리즘과 내림 알고리즘은 색깔의 순서로 작성하거나 실험적으로 합쳐서 계단 보행 알고리즘으로 작성한다.

* 제 2코스:



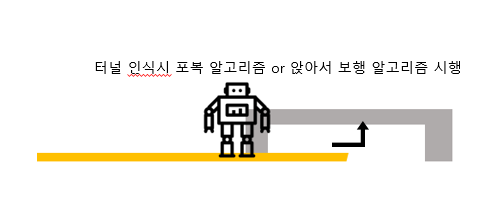
: 외나무 다리 알고리즘은 실험적으로 로봇이 다리를 인식 시 측면으로 보행할지 정면으로 보행할지 정한다. 측면 보행 시 정면으로의 흔들림을 주의하여 모터의 토크를 강화 할 것도 고려한다.

* 제 3코스

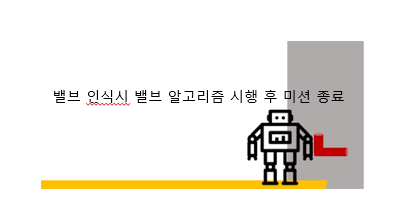


: 커브 길은 라인 추적 알고리즘을 통해 회전 알고리즘을 시행하도록 한다. 문 의 색깔로 문을 인식하고 문 열기 알고리즘을 시행한다. 문 열기 알고리즘은 실 험적으로 문을 미는 방법을 구상한다. 또한 문을 밀 경우 로봇이 틀어지는 것을 고려한다.

* 제 4코스



: 터널은 색깔 또는 터널의 모양으로 인식한다. 터널 인식 시 엎드려서 터널을 지나는 포복 알고리즘을 시행할지 앉아서 보행 알고리즘을 시행할지는 실험적으로 판단한다. 포복 알고리즘을 작성할 시 모션 캡쳐를 이용한다.

* 제 5코스
* 

: 밸브는 빨간색의 비율 또는 모양으로 판단한다. 밸브 알고리즘은 밸브 타워를 받은 후 그에 맞게 모션 캡쳐로 작성한다.

1. **예상되는 장애요인과 해결방안**

**3.1. 프로그래밍시 장애 및 해결**

3.1.1. 빨간색 범위 제한의 문제

: 장애물-콜라캔, 빨간색 계단, 레버의 색깔이 동일한 빨간색으로 인식 될 경우 인식된 범 위로 판단하도록 설정해야 한다. 하지만 이 경우 오차가 발생할 가능성이 크므로 레버는 가능하다면 모양으로 인식하도록 한다. 또는 빨간색-파란색이 나오는 것을 계단으로 인식 하도록 설정 할 수도 있다.

3.1.2. 다리 인식의 문제

: 외나무 다리의 스타트 라인이 없으므로 다른 길과 라인 추적만으로 구별할 수 없다. 유일하게 다리 코스에서 길의 폭이 좁아지며 양쪽에 검은색이 생기므로 이 검은색을 인식하도록 설정한다. 또는 이가 힘들 경우 계단 보행 후는 커브 길이 나 올 때까지 측면 보행을 하도록 한다.

3.1.3. 터널 인식의 문제

; 터널의 색깔이 주변 코스와 비슷하다는 가정하에 터널을 인식하기가 어렵다. 따라서 터 널 상단의 모양을 인식하도록 프로그래밍한다. 또는 모든 라인 추적 과정에서 터널로 판 단 하기 위한 알고리즘을 작성한다. 하지만 후자의 방법의 경우 너무 오랜 시간이 걸리 므로 지양하도록 한다.

**3.2. 수행 중 장애 및 해결**

3.2.1. 라인이탈의 문제

: 라인 추적 알고리즘을 따라 라인을 재 탐색한다. 또한 라인을 카메라 중앙에 위치하도 록 구동체의 위치와 방향을 조정하도록 알고리즘을 작성한다.

3.2.2. 넘어짐의 문제

: 2축 기울기 센서와 2축 자이로 센서가 구동체에 내장되 있으므로 일어나는 알고리즘을 기존의 소스에서 사용한다.

3.2.3. 경기장의 채광 조건 및 환경 고려

: 경기 당시에 리허설을 통해 색의 설정 값을 수정하는 과정을 갖는다.

1. **영상처리 기술 공부 내용**

링크: https://youtu.be/y71ulVj5Drw

1. **단계별 개발계획 및 참여인원과 업무 분담**

:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 6월 | 7월 | 8월 | 9월 | 10월 | 11월 |
| 사전 조사 및 이론 이해 | O |  |  |  |  |  |
| 로보 베이직 학습 |  | O | O |  |  |  |
| 영상 처리 학습 및 알고리즘 작성 |  |  | O | O |  |  |
| 통신 및 동작 구현 |  |  | O | O |  |  |
| 미션 수행 실험 및 오류 개선 |  |  |  | O | O | O |

|  |  |
| --- | --- |
| 인원 | 업무 분담 |
| 전석모 | 로보 베이직 학습, 영상 처리, 알고리즘 작성, 동작 구현 등 |
| 정재호 | 로보 베이직 학습, 영상 처리, 알고리즘 작성, 동작 구현 등 |