차 례

학습모듈의 개요 ---------- 1

학습 1. 인지지능 소프트웨어 개발하기

1-1. 인지지능 소프트웨어 개발 ---------- 3

- 교수·학습 방법 ---------- 10
- 평가 ---------- 11

학습 2. HRI지능 소프트웨어 개발하기

2-1. HRI지능 소프트웨어 개발 ---------- 14

- 교수·학습 방법 ---------- 26
- 평가 ---------- 27

참고 자료 ---------- 29

로봇 모션 제어 및 지능 소프트웨어 개발 학습모듈의 개요

학습모듈의 목표

로봇의 모션 제어 목표 성능과 신뢰성을 만족하는 소프트웨어를 개발할 수 있으며, 로봇이 작업을 수행하거나, 이동하거나, 주변 환경을 인지하거나, 사람과 상호작용하기 위한 지능 기반 소프트웨어를 개발할 수 있다.

선수학습

C 프로그래밍, 로봇 소프트웨어 플랫폼(ROS) 프로그래밍, 로봇 공학, 모션 컨트롤러

학습모듈의 내용체계

학습	학습 내용	NCS 능력단위 요소	
		코드번호	요소 명칭
1. 인지지능 소프트웨어 개발하기	1-1. 인지지능 소프트웨어 개발	1903080320_16v2.1	인지알고리즘 설계하기
2. HRI지능 소프트웨어 개발하기	2-1. HRI지능 소프트웨어 개발	1903080320_16v2.2	HRI지능 소프트웨어 개발하기

핵심 용어

지능로봇, 소프트웨어, 알고리즘, 인공지능, 작업지능, 이동지능, 인지지능, HRI지능

학습 1 인지지능 소프트웨어 개발하기

학습 2 HRI지능 소프트웨어 개발하기

1-1. 인지지능 소프트웨어 개발

학습 목표

- 사용자 요구 분석 결과에 따라 물체, 사람, 환경을 인지하기 위해 필요한 로봇 인지지능 소프트웨어의 목표 사양을 선정할 수 있다.
- 소프트웨어의 목표 사양을 만족하는 인지지능 분석 및 프로그램 구조를 설계할 수 있다.
- 로봇 인지지능을 구현하는 알고리즘을 도출할 수 있다.
- 로봇 인지지능 알고리즘을 기반으로 한 프로그램을 작성할 수 있다.

필요 지식 /

[1] 인지지능

인간의 다섯 가지 감각(시각, 청각, 후각, 미각, 촉각)기관 중 시각 기관은 방대한 정보를 정확하고 빠르게 획득함으로써 인간의 생존과 지능적 활동에 매우 중요한 역할을 한다.

로봇이 센서로 물체와 사람, 환경을 정확히 인지(perception)하기 위해서는 머신비전(machine vision) 기술을 기반으로 물체 인식 기술, 동적인 물체의 시각적 추적, 얼굴 인식, 상황/맥락 이해 기술 등이 필요하다.

1. 머신비전

머신비전은 카메라와 조명 등을 이용해서 피사체의 영상 데이터를 획득하고, 이를 프레임 그래버(frame grabber), 마이크로 컨트롤러, 전용 SoC등으로 처리, 분석 및 해석함으로써 유용한 정보를 얻거나 대상의 3차원 모델을 추출하는 작업 등을 하는 것을 통칭한다.

머신비전은 일반적으로 영상 획득(image acquisition), 영상 처리(image processing), 영상 분석(image analysis)의 과정으로 이뤄진다.

(1) 영상 획득

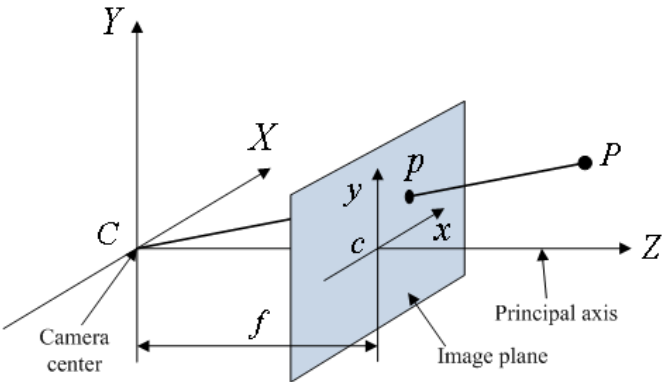
기본적인 시각 센서인 카메라의 핵심 요소인 영상 센서(image sensor)를 이용해서 3차원 세계의 정보를 2차원 영상 평면에 투영하고 매핑하는 것을 영상 획득이라 한다. 가장 대표적으로 사용되는 영상 센서인 CCD(charge coupled device)는 각 화소(pixel)에

입사 된 빛의 강도를 검출하고 이를 전기신호로 변환한다. 프레임 그래버에서는 이 아날로그 신호를 입력받아 A/D(analog to digital) 변환기에서 표본화(sampling)와 양자화(quantization) 과정을 거쳐 디지털 값으로 변환한다.

표본화란 매 샘플 시간에 검출한 아날로그 신호를 다음 샘플 시간까지 고정하는 것을 의미하고, 양자화란 표본화된 아날로그 값을 디지털 값으로 변환하는 것으로 A/D변환기의 비트수 N에 따라 해상도(resolution)가 2^N으로 증가한다. 양자화의 해상도는 깊이 해상도(depth resolution)라고도 불리며 흑백 영상은 1비트로 양자화한 경우에 해당된다.

[그림 1-1]은 대표적으로 많이 사용되는 핀홀 카메라(pinhole camera)의 기하학적 관계를 나타낸 그림이다. 실제 3차원 공간상에 있는 한 점 $P=(p_X, p_Y, p_Z)$는 이 점에서 카메라 중심점 C까지 이은 직선이 2차원 영상 평면에서 만나는 점인 화소 $p=(p_x, p_y, p_z)$에 투영된다. [그림 1-1]에서 카메라 중심점과 영상 평면 간의 거리를 f 라고 하면, 닮은 삼각형의 비례식을 이용하여 다음의 관계를 유도할 수 있다.

$$p_x = f\frac{p_X}{p_Z},\ \ p_y = f\frac{p_Y}{p_Z},\ \ p_z = f \qquad \text{식 (20)}$$



출처: Hartley and Zisserman(2003). Multiple View Geometry. Cambridge University Press.
[그림 1-1] 핀홀 카메라 모델

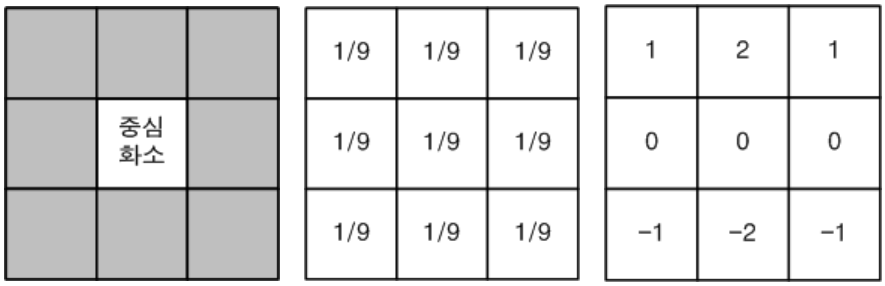
i와 j가 음이 아닌 정수일 때 영상 평면에서 화소의 위치를 나타내는 인덱스는 (i, j)로 표현된다. 일반적으로 기준 화소는 영상 평면의 좌측 최상단에 위치하며 그 인덱스는 (0, 0)이다. 화소가 영상 평면의 아래쪽으로 갈수록 i값이 증가하고, 오른쪽으로 갈수록 j값이 증가한다.

(2) 영상 처리

영상 처리는 한 영상을 다른 영상으로 변환하는 기술로서 로봇에 필요한 기술에는 잡음 제거, 경계 검출, 영상의 이진화 등이 있다.

잡음 제거는 영상의 화소 중에서 잡음(noise)이 발생한 화소의 디지털 값을 마스크

(mask)를 이용해서 주변 화소값들과 평균함으로써 비슷하게 만들어 주는 작업이다. 일반적으로 [그림 1-2]에 보인 8-이웃(8-neighborhood) 화소법을 많이 사용하는데, 3×3 크기의 마스크를 이용해서 현재 화소(중심 화소)와 인접한 팔방의 화소 값들을 읽어 들이고 여기에 마스크의 가중치를 곱하여 더함으로써 평균을 계산하는 방식을 취한다.
경계 검출은 물체나 얼굴 인식을 위해 기본적으로 필요한 작업으로서, [그림 1-2]에 보인 소벨(Sobel) 마스크와 같은 것을 사용하여 현재 화소에서 소벨마스크로 가중 평균한 값과 이전 위치 화소에서 소벨마스크로 가중 평균한 값과의 변화량 크기가 임계치(threshold) 이상인 화소를 경계 화소라고 판단한다(정성환 · 이문호, 2008).



[그림 1-2] 8-이웃 화소법과 산술평균형 마스크, 소벨마스크(수평 경계 검출용)

영상의 이진화는 물체와 배경만으로 구성 된 단순한 영상의 경우, 임계치를 이용해서 배경에 해당되는 화소에는 0, 물체에 해당되는 화소에는 1의 값을 할당하는 기법이다. 실제 영상의 경우 이진화 임계치를 정하기 위해 가로축은 화소의 밝기값(명암), 세로축은 해당 명암값을 가진 화소의 개수(빈도)를 나타내는 히스토그램(histogram)을 그려서 빈도의 형상이 계곡 모양을 이루는 영역의 중간 지점 명암값을 찾아서 임계치로 선택한다.

(3) 영상 분석

영상 분석은 영상 처리의 결과물 영상을 이용해서 영상의 정보를 얻어내는 것으로, 영역 분할(segmentation)과 특징 추출(feature extraction) 작업이 있다.
영역 분할은 영상 내에서 비슷한 특징을 가지는 이웃 화소들을 영역별 임계치와 경계 검출을 통해 하나의 영역으로 분리하여 처리하는 것이다. 영역 분할은 관심 영역(region of interest)을 정하고 영역별로 고속으로 분석하는 데 유용하게 사용된다.
특징 추출은 영상에 있는 물체의 기하학적 특징(평균 밝기, 최대/최소 밝기, 면적, 둘레, 길이, 폭, 지름, 종횡비 등)을 검출하는 기술이다.

2. 물체 인식

물체 인식(object recognition)이란 로봇에 있는 카메라, 스테레오 비전, 같은 3차원 센서로 입력 받은 영상 데이터를 이용해서 어떤 물체가 어느 위치에 어떤 자세로 있는지를 알고리즘으로 계산하는 것을 의미한다.

강인한 물체 인식이 가능하기 위해서는 다양한 물체들이 놓여 있을 때 잡음이 있어도 물체들을 잘 인식할 수 있어야 하며, 물체가 가려지거나 틀어져 있을 때에도 잘 인식할 수 있어야 한다.

물체 인식에는 원형 정합(template matching)과 특징 정합(feature matching) 기술들이 개발되어 왔으며, 최근에는 DNN의 눈부신 발전으로 일부분에서 사람보다 뛰어난 영상 인식 성공률을 보이고 있다.

(1) 원형 정합

원형 정합은 영상에서 분석된 물체를 미리 저장되어 있는 표준 모델인 원형과 비교하여 그중에서 가장 가까운 모델을 인식 물체로 선정하는 방법이다.

(2) 특징 정합

특징 정합은 영상에서 분석된 물체의 특징점 또는 차별성 있는 부분적 구조를 부호화(encode)하여 다른 영상에 있는 특징점이나 부분적 구조와의 정합성을 판별하여 물체를 인식하는 기술이다.

물체의 병진(translation), 회전, 배율(scale) 등의 변화와 조명, 잡음, 양자화와 같은 변화가 일어나도 변함없이 물체의 특징을 잘 추출하기 위해 특징 정합은 물체의 코너나 경계선과 같이 구분 가능성이 높은 곳에서 실행되어야 한다.

경계선 검출을 위해 많이 사용되는 방법은 캐니(CANNY) 연산자(CANNY, 1986)와 영상 잡음 기반 경계 검출기이며, 코너 검출을 위해서는 해리스(HARRIS) 코너 검출기가 많이 사용된다.

영상의 경계가 원(blob)으로 피팅(fitting)되는 영역을 찾아내는 영역 검출기로는 가우시안(gaussian) 분포를 2차 미분하여 얻은 LoG(laplacian-of-gaussian)를 근사화한 함수인 DoG(difference-of-gaussian)를 많이 사용한다.

3. 물체의 시각적 추적

로봇이 인지지능을 가지기 위해서는 동적이고 복잡한 환경에서 사람이나 물체, 동작을 실시간으로 잘 인식하고 이를 연속적으로 추적하는 기술이 중요하다. 이를 위해 카메라가 움직이는 상황에서 단일 또는 다중의 물체 또는 사람을 추적하는 기술과, 2차원의 동작 및 자세 영상을 3차원으로 확장했을 때의 동작 및 자세 등을 추정하는 기술 들이 필요하다.

2차원 영상에서의 다중 물체 추적 기술에는 가려짐이 발생할 수 있는 상황에서 안정적으로 추적할 수 있게 하기 위해 칼만필터, 파티클필터가 많이 사용된다.

4. 얼굴 인식

로봇이 사람과 공존하기 위해서 중요한 기술 중의 하나가 상대방의 얼굴을 인식해서 로봇의 소유자인지, 로봇이나 소유자와 관계된 사람인지, 낯선 사람인지를 기본적으로 구별할 수 있어야 한다.

5. 상황 인식 기술

상황 인식(context awareness)이란 로봇이 물체와 사람, 환경 등을 개별적으로 인식한 후 이미 알고 있는 상징적 지식과 연계하여 물체 간, 물체와 사람 간, 또는 사람과 사람 간의 기하학적, 물리적, 또는 사회적인 관계를 인식하는 기술로서 인간에게 상황에 따라 적절한 반응이나 서비스를 예측하여 사람에게 수행하기 위해 필수적인 기술이다.

상황 인식 기술을 위해서는 다음의 기술이 필요하다.

(1) 감지된 상황을 기호로 표현하는 기술

(2) 상황을 표현하는 온톨로지(ontology)

(3) 인식된 물체들의 관계를 인식하고 이로부터 의미를 유추하는 기술

수행 내용 / 인지지능 소프트웨어 개발하기

재료 · 자료

- 사용자 요구 분석서
- 프로그램 구조 설계서
- 알고리즘 기술서
- 프로그램 코드

기기(장비 · 공구)

- 컴퓨터, 프린터
- 프로그램 개발용 소프트웨어
- 문서 작성용 소프트웨어

안전 · 유의 사항

- 인지지능을 구현하기 위해 사람과 로봇이 가까이 있는 경우 로봇의 팔이나 기구부가 사람에게 부딪치지 않도록 안전거리를 유지해야 한다.
- 인지지능 소프트웨어를 다운로드해서 사용할 경우 라이선스를 확인해서 무료로 사용할 수 있는지 여부를 확인해야 한다.

수행 순서

1 사용자 요구 분석 결과에 따라 물체, 사람, 환경을 인지하기 위한 로봇 인지지능 소프트웨어의 목표 사양을 선정한다.

1. 사용자 요구 분석서와 수행할 인지지능 태스크를 분석하여 물체/사람/환경 인식의 인식 속도와 정확성, 반복성, 강인성(robustness) 등의 항목으로 구성 된 인지지능 기능 요구서를 작성한다.
2. 인지지능을 수행할 로봇 타입과 인지 대상(물체, 사람, 환경)에 따라 적절한 카메라와 센서, 영상 인식 알고리즘과 소프트웨어를 검색하고 선택한다.
   (1) 물체 인식을 위해 적절한 카메라를 선택하고, 적용 가능한 알고리즘을 학습하고, 관련 프로그램을 확보하여 분석한다.
   (2) 사람 인식을 위해 적절한 카메라를 선택하고, 적용 가능한 알고리즘을 학습하고, 관련 프로그램을 확보하여 분석한다.
   (3) 환경 인식을 위해 적절한 카메라를 선택하고, 적용 가능한 알고리즘을 학습하고, 관련 프로그램을 확보하여 분석한다.
3. 선택한 로봇 타입, 카메라, 영상 인식 알고리즘에 대해 소프트웨어로 구현 가능한 물체/사람/환경 인식 속도와 정확성, 강인성 값을 목표로 설정한다.

2 인지지능 소프트웨어의 목표 사양을 만족하는 기본 지능을 분석하고 프로그램 구조를 설계한다.

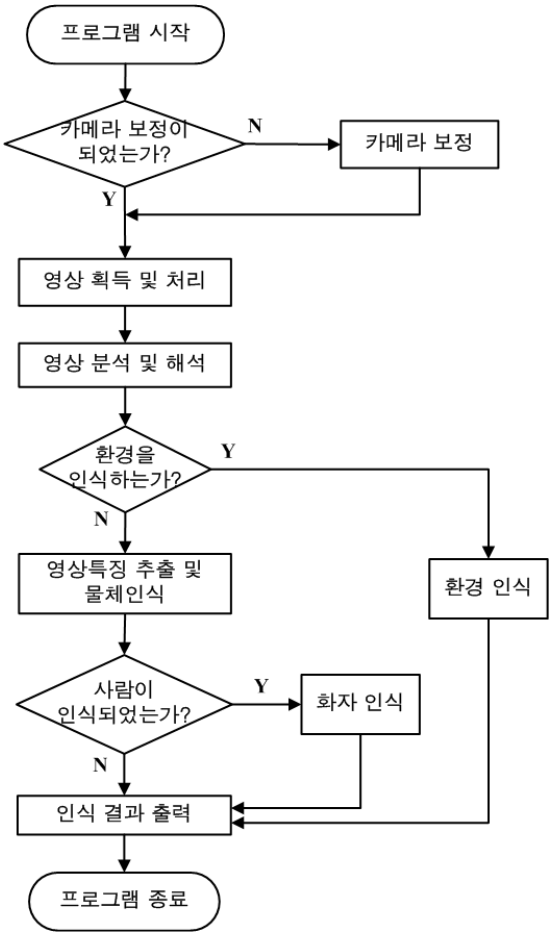
1. 인지 대상별로 요구되는 인지 기능의 특성과 조건을 분석하고 적절한 알고리즘을 결정한다.
   (1) 인지해야 할 대상이 물체이면 정적(static)이면서 가려짐(occlusion)에 강인한 SIFT(scale invariant feature transform) 알고리즘이 적절하다.
   (2) 인지해야 할 대상이 사람이면 동적으로 움직이는 물체를 잘 검출하며 얼굴과 신체의 특징점을 잘 검출하는 SIFT 알고리즘이 많이 사용된다.
   (3) 인지해야 할 대상이 환경이면 구조적인 실내인지 비구조적인 실외인지를 구분하고 복잡한 배경(background clutter)에 효율적인 SIFT 알고리즘을 사용한다.
   (4) 인식 속도가 빨라야 한다면 SURF(speed-up robust features)를 사용하고, 느려도 된다면 SIFT를 사용한다.
2. 수행해야 할 인지 기능에 요구되는 지능을 분석한다.
   (1) ALFUS를 참고하여 수행해야 할 작업의 복잡도, 환경의 복잡도, 인간-로봇 상호작용 정도 별로 자율화 지수가 0~10 중에서 어느 정도인지를 파악한다.
   (2) 전체 자율화 지수가 0에 가까우면 사람이 로봇의 영상 데이터를 보고 수동으로 물

체를 인식하여 라벨을 붙이는 것이고, 자율화 지수가 10에 가까우면 DNN을 이용해서 로봇이 자율적으로 물체를 인식할 수 있음을 의미한다.

3. 인지지능 소프트웨어의 프로그램 구조를 설계한다.

(1) 인지지능 소프트웨어의 일반적인 프로그램 구조는 [그림 1-3]과 같다.

(2) 인지지능을 적용하기 전에 로봇의 카메라에 보정(calibration) 작업이 시행되었는지를 확인해야 한다. 카메라 보정이란 피사체가 있는 3차원 공간 좌표계와 2차원 영상 좌표계 사이의 기하학적 관계를 수식으로 표현하는 작업을 의미한다. 이를 위해서 카메라의 내부 변수(카메라의 광학, 기하학 및 영상 센서의 특성을 정의하는 변수)와 외부 변수(카메라의 회전각과 이동 변위) 값을 조정해야 한다.



[그림 1-3] 인지지능 소프트웨어의 프로그램 흐름도

3 로봇 인지지능을 구현하는 알고리즘을 도출하고 이를 기반으로 한 프로그램을 작성한다.

학습 2 HRI지능 소프트웨어 개발하기

2-1. HRI지능 소프트웨어 개발

학습 목표

- 사용자 요구 분석 결과에 따라 로봇이 사람과 상호작용을 하기 위해 필요한 HRI지능 소프트웨어의 목표 사양을 선정할 수 있다.
- 소프트웨어의 목표 사양을 만족하는 HRI지능 분석 및 프로그램 구조를 설계할 수 있다.
- HRI지능을 구현하는 알고리즘을 도출할 수 있다.
- HRI지능 알고리즘을 기반으로 한 프로그램을 작성할 수 있다.

필요 지식 /

[1] HRI지능

일반적으로 산업용 로봇은 제자리에서 무겁고 큰 물건을 들어 올리거나 옮기기 때문에 안전을 위해 로봇의 작업 공간(working space) 안에 작업자나 일반인이 들어오지 못하도록 로봇 주변에 안전 펜스를 설치한다. 그러나 서비스 로봇처럼 사람 가까이에서 다양한 서비스를 제공하거나 스마트 팩토리(smart factory)용 로봇처럼 사람과 함께 작업을 하는 경우 작업의 정확성과 안전뿐만 아니라 작업자와의 효율적이면서도 감성적인 상호작용 기술 또한 필요하다. 이를 위해 최근 HRI(human-robot interaction) 분야의 기술의 중요성이 커지고 있는데, HRI는 '로봇이 사용자(인간)의 의도를 파악하고, 그에 따라 적절하게 반응하고 행동함으로써 상호간에 의사소통하고 협력할 수 있게 하는 기술'로 정의할 수 있다.

HRI 기술의 목표는 다음과 같이 크게 두 가지로 구분할 수 있다.

- 사람과 로봇 사이에 보다 자연스럽고 효과적인 상호작용이 가능하게 하는 알고리즘 개발
- 사람이 로봇과의 상호작용에서 기대하는 모델을 정의하여 로봇 설계에 반영

HRI는 인간-컴퓨터 상호작용(human-computer interaction), 인공지능, 로봇 공학, 자연어 이해, 사회과학(social science) 등의 분야와 밀접하게 연관되어 있다.

이상무(2011)에 의하면 HRI의 핵심 기술은 인식(perception), 판단(cognition), 표현(expression) 기술로 요약될 수 있다.

1. 인식 기술

인식 기술은 로봇이 각종 센서 정보로부터 사용자의 표정이나 동작(제스처)을 인식함으로써 사용자의 의도나 반응을 파악하거나, 센서 정보와 지식을 활용하여 현재 처한 상황(context)을 파악할 수 있게 하는 기술이다. 이를 위해 카메라를 이용해서 물체, 화자, 표정, 동작 등을 인식하거나 사람을 추종하는 기술, 마이크를 이용해서 음원, 음색, 음성을 인식하는 기술, 압력 센서나 접촉 센서 등을 이용해서 사용자의 접촉 여부를 인식하는 기술 등이 요구된다.

(1) 휴먼인식

휴먼인식은 로봇이 카메라 영상으로부터 사람을 인식하는 기술로서 사람이 근거리(0.5~2.5 m)에 서 있는 경우 정면과 측면의 얼굴을 검출하고, 중거리(2~5 m)에 서 있을 경우에는 얼굴과 어깨를 검출하고(오메가 검출기), 원거리(4~9 m)에 있는 경우에는 전신을 검출한다.

휴먼인식 결과로서 새로운 사람이 근거리에 있는 경우 영상 인식과 음성 인식을 결합하여 화자 인식을 시도하고, 중거리 및 근거리에 사람이 있는 경우 제스처 인식 기술을 결합하여 적합한 이동(접근 또는 추종) 기능을 실행한다.

출처: 이상무(2011). 인간로봇상호작용(HRI) 기술의 현황과 발전 방향. p.8.
[그림 2-1] ETRI에서 개발한 얼굴과 상반신, 전신을 검출하는 휴먼인식 소프트웨어 실행 화면

(2) 동작인식

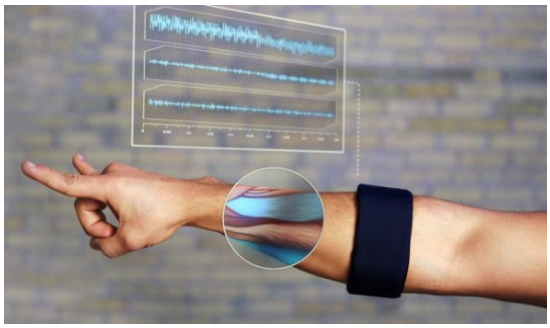
동작인식은 사람이 음성 대신 전신, 다리, 팔, 손, 손가락 등을 이용해서 로봇에게 특정 명령이나 의도를 전달하는 기술로 스마트 인터랙션(smart interaction) 기술과 밀접한 관련이 있다.

동작인식은 장치를 이용해서 얻은 데이터를 사용하는 접촉식과 주로 카메라를 이용해서 얻은 동작 데이터를 사용하는 비접촉식으로 구분된다.

동작인식 센서로 많이 사용되는 것으로 접촉식에 해당하는 탈믹랩(THALMIC LABS)의 마이요(MYO)와 비접촉식에 해당하는 마이크로소프트의 키넥트(KINECT), 립모션 사의 립모션(LEAP MOTION)을 꼽을 수 있다.

[그림 2-2]에 나타낸 마이요는 팔에 착용하는 암밴드(arm-band) 형태의 입력 장치로 근육 센서와 6축 가속도 센서로 근육의 움직임을 인식하여 손가락과 팔의 25가지 동작을 인식한다. [그림 2-3]에 나타낸 키넥트는 RGB센서와 IR(적외선)센서, 다중배열 마

이크로폰을 이용하여 사람의 신체 부위를 검출하고 객체의 원근을 구별한다. 립모션은 2개의 IR 카메라와 3개의 IR LED의 조합으로 근거리에서 손가락의 위치를 인식하여 컴퓨터에 입력하는 인터페이스이다.



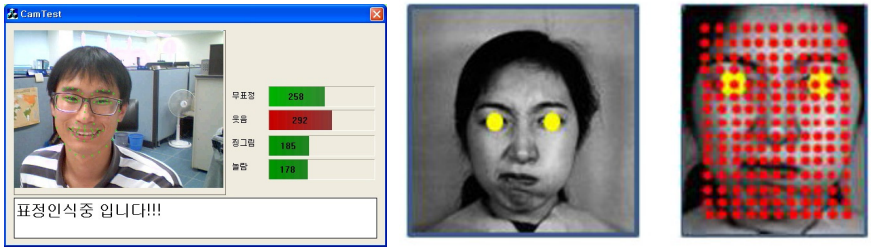
출처: John Hewitt(2013.02.27.). The MYO gesture-control armband senses your muscle's movements. http://www.extremetech.com/extreme/149335-the-myo-gesture-control-armband-senses-your-muscles-movements에서 2016.10.03. 검색.
[그림 2-2] 동작인식 센서인 마이요

[그림 2-3] 키넥트 센서

(3) 감정인식

감정인식은 로봇이 사람의 내적인 정서적·심리적 상태를 인식하여 적절한 정서적 반응을 하거나, 사람과 함께 작업하는 경우 작업 강도를 조절하는 데 필요한 기술이다. 일반적으로 정면 얼굴의 각 특징점(눈, 코, 입술)의 위치 변화를 분석하여 중립적 감정(무표정, 놀람), 긍정적인 감정(평안, 행복, 즐거움)과 부정적인 감정(슬픔, 화남)을 구분하여 인식한다.

일례로 숭실대학교에서는 가버 웨이브렛(GABOR WAVELET) 변환을 통해 화남, 행복, 평온, 슬픔, 놀람의 5가지 표정을 평균 10.25 m초의 수행 시간과 87~93%의 성능으로 인식했다.



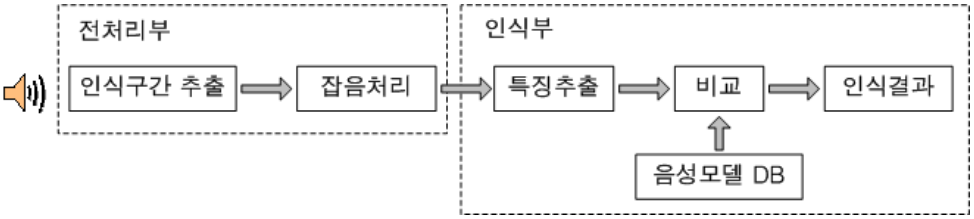
출처: 이상무(2011). 인간로봇상호작용(HRI) 기술의 현황과 발전 방향, 9-10쪽
[그림 2-4] 표정인식 기술의 구현 화면

(4) 음성인식

음성인식은 사람의 말을 인식해 텍스트로 바꿔주거나 해당 명령을 수행하는 기술로서 1950년대에 등장해 다음의 단계를 거치며 모든 사용자를 대상으로 보다 많은 어휘, 자연스러운 대화체를 인식하며, 인식률을 높이는 방향으로 발전해 가고 있다.

- 음절·음소 인식
- 고립 단어 인식
- HMM
- 신경회로망 기반 인식 기술
- n-gram 기반의 언어 모델 기술법
- 스핑크스(SPHINX) 시스템(자연언어와 불특정화자 연속 음성인식)
- 배경 잡음과 반향 등에도 강인한 음성인식

음성인식 과정은 [그림 2-5]와 같이 크게 전처리부와 후처리부로 나뉜다.



[그림 2-5] 음성인식의 전처리부와 후처리부

음성인식 분야에서는 애플의 시리(SIRI)와 구글 나우(GOOGLE NOW)가 기술을 선도하고 있으며 아마존의 알렉사(ALEXA)와 마이크로소프트의 코타나(CORTANA)가 음성 인식 서비스 경쟁에 합류했다.

IBM의 인공지능인 왓슨(WATSON)은 음성을 인식하고 해석하며 합성하는 자연어 처리(natural language processing)를 할 수 있으며, 아마존에서 출시한 에코(ECHO) 또한 음성인식만으로 동작한다.

오픈소스 음성인식 기술로서 GOOGLE에서는 음성을 문자 포맷으로 변환하는 STT(speech to text) Web Speech API를 2013년 발표했으며, 이후에 APPLE의 SIRI와 유사한 기능을 하는

Google Voice Search Hotword를 발표했다. 그 외의 유명한 오픈소스 음성인식 엔진들로는 CMU의 SPHINX그룹과 SUN MICROSYSTEMS 연구소, MERL(mitsubishi electric research labs), HP 등이 협력하여 연구 개발 중인 음성엔진인 CMUSPhinx(https://github.com/tilo/cmusphinx-1), MIT에서 개발 중이며 iPhone, iPod Touch 모바일 웹브라우저에서 음성인식을 하는 toolkit인 WAMI (http://www.csail.mit.edu/research/playground/wami) 등이 있다.

2. 판단 기술

판단 기술은 로봇이 인식 기술로 얻은 환경 및 사용자 관련 정보들로부터 인터랙션 모델, 태스크 모델, 온톨로지(ontology) 등을 기반으로 상황인식(context awareness)을 수행함으로써 현재의 상황과 사용자의 의도를 로봇이 이해 가능한 정보들로 변환하는 기술을 의미한다.

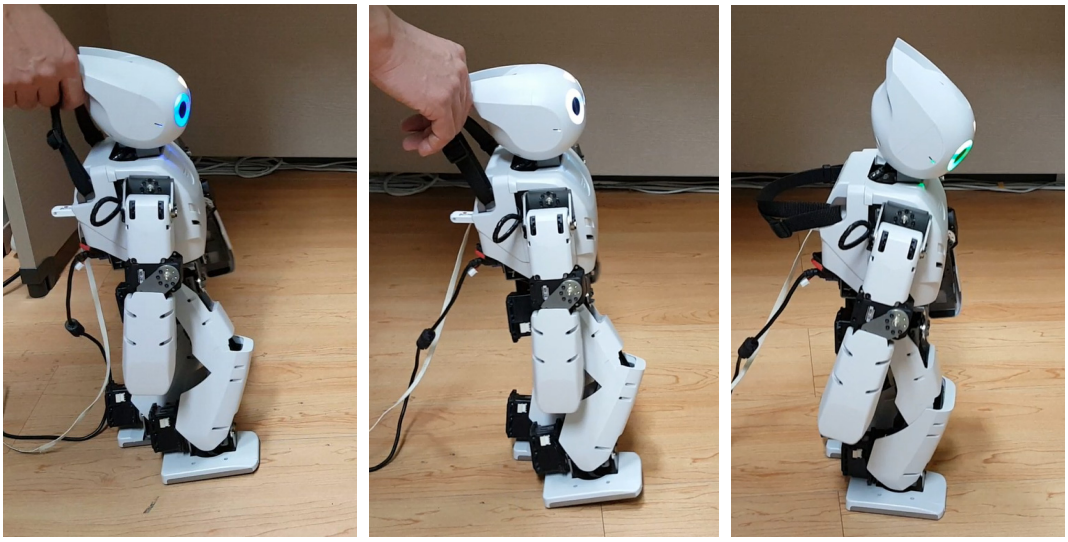
상황인식은 다양한 종류의 센서와 상태 데이터들을 종합적으로 고려하고 필요 시 메모리나 데이터베이스에 저장되어 있는 정보들을 활용하여 추론하고 학습해야 하므로 Soar 같은 인지 에이전트 아키텍처를 사용할 필요가 있다.

3. 표현 기술

표현 기술은 앞서 설명한 인식-판단 과정을 통해 로봇이 사용자에 대해 정보를 획득하고 상황을 판단하고 나서 이에 따라 액추에이터나 장치들을 이용하여 적절히 로봇의 의도나 계획, 감정을 사람에게 표현하는 기술을 의미한다.

표현 기술에는 로봇이 바퀴나 다리(2족, 4족, 6족 이상)를 이용해서 사용자에게 다가가거나 사용자를 추종하는 주행(navigation), 로봇의 얼굴에 있는 LED나 디스플레이 또는 제스처를 통한 감정 생성 및 표현, TTS(text to speech) 소프트웨어를 이용한 음성합성 기술 등이 이에 해당된다.

[그림 2-6]은 인간의 보행 패턴을 분석한 결과를 토대로 휴머노이드 로봇(ROBOTIS-OP)이 감정을 가진 상태로 걸어가는 것처럼 보이는 패턴을 실험으로 구현한 것이다. 첫 번째는 아무런 감정을 가지고 있지 않은 상태로 걸어가는 표준 보행 패턴이며, 눈 LED의 색깔은 푸른색으로 설정되어 있다. 두 번째는 즐거운 상태에서 걸어가는 보행 패턴으로서 상체가 약간 뒤로 젖혀지고 눈 LED의 색깔은 흰색으로 바뀌었으며, 걷는 속도와 보폭이 증가했다. 세 번째는 슬픈 상태에서 걸어가는 보행 패턴을 나타내는 것으로, 상체가 약간 앞으로 숙여지고 눈 LED의 색깔은 녹색으로 바뀌었으며 걷는 속도와 보폭이 표준 보행 패턴에 비해 감소한 특징을 보인다. 이렇게 휴머노이드 로봇의 보행 패턴을 바꾸면 보는 사람은 휴머노이드 로봇이 표현하고자 하는 감정을 직관적으로 알 수 있다.



[그림 2-6] 휴머노이드 로봇의 감정상태에 따른 보행 패턴(왼쪽부터 평상시, 즐거울 때, 슬플 때를 표현)

이와 같이 로봇이 다양한 매개체를 이용해서 감정이나 의사를 표현하는 것을 다중 모드(multimodal) 표현이라고 한다.

(1) 음성 합성

음성합성은 음파를 컴퓨터가 자동으로 만들어 내는 기술로서, 모델로 선정된 사람의 말소리를 녹음하여 일정한 음성 단위로 분할한 다음, 라벨을 붙여 합성기에 입력하였다가 입력된 텍스트에 맞춰 필요한 음성 단위만을 다시 합성하여 말소리를 만들어내는 기술이다.

리눅스 Ubuntu 환경에서 음성 합성을 위한 대표적인 TTS 오픈소스로는 영국 에딘버러 대학에서 개발한 Festival Speech Synthesis System(http://www.cstr.ed.ac.uk/projects/festival/)과 구글 안드로이드 플랫폼에서 채택된 SVOX(현재 NUANCE사에 합병)의 Pico(http://elinux.org/RPi_Text_to_Speech_(Speech_Synthesis)) 등이 있다.

수행 내용 / HRI지능 소프트웨어 개발하기

재료・자료

- 사용자 요구 분석서
- 프로그램 구조 설계서
- 알고리즘 기술서
- 프로그램 코드

기기(장비・공구)

- 컴퓨터, 프린터
- 프로그램 개발용 소프트웨어
- 문서 작성용 소프트웨어

안전・유의 사항

- 로봇의 종류는 상당히 다양하므로 사용자 요구 분석서를 잘 숙지하여 구현하고자 하는 HRI 기능을 수행하기에 적절한 로봇 타입을 선정할 필요가 있다.
- HRI를 구현할 때 로봇의 팔이나 기구부가 사람에게 부딪쳐서 상해를 가하지 않도록 안전거리를 유지해야 한다.
- HRI 소프트웨어를 다운로드해서 사용할 경우 라이선스를 확인해서 무료로 사용할 수 있는지 여부를 확인해야 한다.

수행 순서

[1] 사용자 요구 분석 결과에 따라 로봇이 사람과 상호작용을 하기 위해 필요한 HRI지능 소프트웨어의 목표 사양을 선정한다.

1. 사용자 요구 분석서와 수행할 HRI태스크를 분석하여 자연스러움(naturalness), 응답성(responsiveness), 사전 행동성(proactiveness) 등의 항목으로 구성 된 HRI 기능 요구서를 작성한다.
2. 상호작용에 필요한 로봇의 HRI 분야(인식, 판단, 표현)별 기능을 알아보고 해당 기능을 구현하기 위해 필요한 센서와 부품, 소프트웨어, 알고리즘을 검색하고 선택한다.
3. HRI를 수행할 로봇 타입을 선택하고 로봇 시뮬레이션 소프트웨어를 다운로드 받는다.

로봇 전용 소프트웨어가 없다면 범용 로봇 시뮬레이터를 다운로드 받거나 직접 코드를 작성한다.

(1) HRI를 수행하기 위해서는 사람이 쉽게 공감할 수 있으며 로봇의 감정과 의사를 표현하는 데 용이한 휴머노이드 로봇 플랫폼이 유리하다.

(2) HRI에 사용될 수 있는 소형 휴머노이드 로봇으로는 로보티즈(Robotis)의 로보티즈-OP, 프랑스 알데바란 로보틱스(Aldebaran Robotics)의 나오(Nao)가 있으며, 키가 1m 이상인 바퀴형 휴머노이드 로봇으로는 로보케어(Robocare)의 실벗(Silbot)과 소프트뱅크(SoftBank)의 페퍼(Pepper) 등이 있다.

4. 선택한 로봇 플랫폼에 대한 HRI지능 소프트웨어의 자연스러움, 응답성, 사전 행동성의 목표값을 설정한다.

2 HRI지능 소프트웨어의 목표 사양을 만족하는 기본 지능을 분석하고 프로그램 구조를 설계한다.

1. HRI의 인식·판단·표현지능별로 사용할 기술과 해당 알고리즘을 결정한다.

(1) HRI의 인식 기술(휴먼인식, 동작 인식, 감정 인식, 음성인식 등) 중 어떤 인식 기술을 사용할지 결정하고 그에 필요한 알고리즘을 학습한다.

(2) HRI의 판단 기술을 위해 어떤 인지 아키텍처(Soar, ACT-R, Icarus 등)를 사용할지 결정하고 그 아키텍처에서 맞게 프로그램을 작성한다.

(3) HRI의 표현 기술(주행, 감정 표현, 음성합성 등)중 어떤 기술을 사용할지 결정하고 그에 필요한 알고리즘을 학습한다.

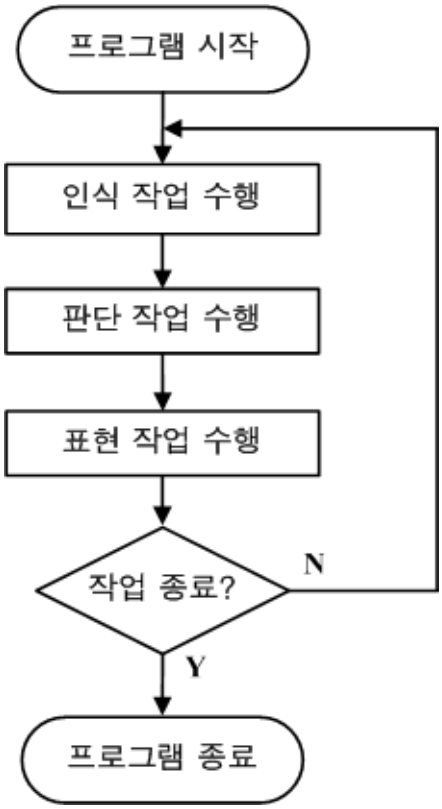
2. 수행해야 할 HRI 기능에 요구되는 지능을 분석한다.

(1) ALFUS를 참고하여 수행해야 할 작업의 복잡도, 환경의 복잡도, 인간-로봇 상호작용 정도 별로 자율화 지수가 0~10 중에서 어느 정도인지를 파악한다.

(2) 전체 자율화 지수가 0에 가까우면 로봇은 높은 HRI 성능을 갖춰서 사람이 지시하는 대로 태스크를 잘 수행해야 하고, 10에 가까울수록 사람과 상호작용 없이 자율적으로 태스크를 수행하며 사람과 대화를 나누거나 감정이나 동작을 적절하게 표현할 수 있다.

3. HRI지능 소프트웨어의 프로그램 구조를 설계한다.

HRI지능 소프트웨어의 일반적인 프로그램 구조는 [그림 2-7]과 같다.



[그림 2-7] HRI지능 소프트웨어의 프로그램 흐름도

3 구현하고자 하는 전체 HRI 태스크의 시나리오를 결정하고, 인식·판단·표현지능별로 필요한 소프트웨어 패키지를 사용하여 프로그램을 작성한다.

1. 시나리오 작성 및 판단 지능 아키텍처 설치

- 간단한 HRI 태스크의 예로서 다음과 같이 휴머노이드 로봇 적용 시나리오를 작성한다.

(인식) 로봇은 "Stand", "Down", "Yes", "No", "Go", "Stop" 의 여섯 가지 기본적인 음성 명령을 인식할 수 있다.

(인식) 로봇은 종이에 그려진 'O', 'X', 'Δ' 의 세 가지 상징을 인식할 수 있다.

(판단) 초기 자세에서 "Stand" 명령으로 로봇이 일어서게 한 후, "Go" 라고 명령하면 앞으로 걸어간다.

만약 로봇이 테이블 끝에 서있는 위험한 상황에 처하거나 사람으로부터 부적절한 명령이 주어진 것을 의미하는 'X' 를 보면 그 자리에 멈추고, 그 다음부터는 "Go" 라고 명령해도 반응하지 않는다.

(표현) 로봇이 "Sorry, I cannot walk." 라고 대답함으로써 부적절한 명령에 대해 거부하고 있음을 사람에게 표현한다.

(인식) 이러한 비정상적인 상태가 종료되었음을 의미하는 'O' 가 그려진 종이를 로봇에게 보여주면 그 다음부터는 사람의 명령대로 따른다.

- 상기 시나리오와 같이 비동기적으로 다중(음성과 영상) 입력이 들어오는 HRI 태스크를 수행하기 위해 병렬적으로 복수의 규칙을 처리할 수 있는 Soar를 인지 에이전트 아키텍처로 사용한다. 이를 위해 아래의 주소에서 SoarSuite 패키지 9.5.0 버전을 다운로드해서 설치한다.

  http://soar.eecs.umich.edu/articles/downloads/soar-suite/219-soar-suite-9-5-0-beta

- Soar의 main 코드(soar.py)를 포함하는 soarwrapper 패키지를 만든다.
- 로보티즈-OP(이하 OP)에 ROS Hydro버전(OP의 컴퓨팅 사양에 적합)을 설치하고 다음의 주소에서 기본 패키지들을 다운로드한다.

```
OP 설명 패키지: https://github.com/ROBOTIS-OP/robotis_op_common
OP 프레임워크 패키지: https://github.com/ROBOTIS-OP/robotis_op_framework
ROS 제어 패키지: https://github.com/ROBOTIS-OP/robotis_op_ros_control
ROS 론치 패키지: https://github.com/ROBOTIS-OP/robotis_op_launch
OP 카메라 패키지: https://github.com/ROBOTIS-OP/robotis_op_camera
```

2. 인식지능 패키지 설치
   - 영상인식을 위해 다음 주소에서 find-object 패키지를 다운로드하고 빌드한다.

     https://github.com/introlab/find-object

   - 음성인식을 위해 다음 주소에서 Pocketsphinx 패키지를 다운로드하고 빌드한다.

     https://github.com/mikeferguson/pocketsphinx

3. 표현지능 패키지 설치
   - 음성합성을 위해 다음 주소에서 sound_play 패키지를 다운로드하고 빌드한다.

     https://github.com/ros-drivers/audio_common.git

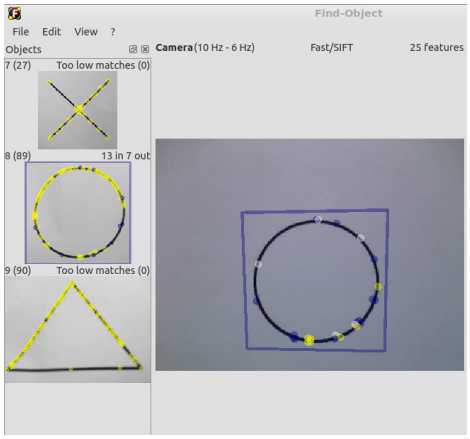
   - 여기서 소개하는 HRI 태스크 실습 패키지는 RoboFriend 웹사이트 (http://www.robofriend.kr)의 자료실에서 다운로드 받을 수 있다.

4. ROS에서 전체 패키지를 실행한다.
   - 터미널을 열고 다음 명령을 실행한다.

```
$ roscore
```

   - 새 터미널을 열고 다음 명령을 실행한다. Find-object GUI가 열리면 'File/Load' 세션으로 가서 "lesson_2.bin" 을 로드한다. 그러면 로봇이 학습한 'O', 'X', 'Δ' 이미지들을 [그림 2-8]과 같이 볼 수 있다.

```
$ rosrun find_object_2d find_object_2d image:=robotis_op/camera/image_raw
```



출처: Find-object GUI. 스크린샷.
[그림 2-8] Find-object GUI에서 'O' 마크를 인식하는 모습

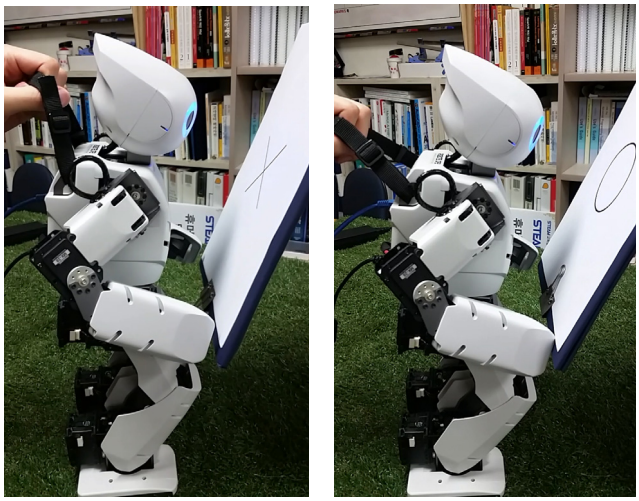
- 새 터미널을 열고 다음과 같이 론치 파일을 실행하면 OP가 음성인식을 시작하며 HRI태스크를 실행할 준비를 한다. 이때 로봇의 파손을 방지하기 위해 OP가 무릎을 가지런히 굽히고 앉아있는 자세가 되도록 조정한다.

```
$ roslaunch robotis_op_onboard_launch robotis_op_whole_robot.launch
```

- 새 터미널을 열고 다음 명령을 실행한다. 여섯 가지 음성 명령("Stand", "Down", "Yes", "No", "Go", "Stop") 중 "Stand" 명령으로 OP를 일으켜 세운 후 나머지 명령을 말하면 OP가 명령을 실행한다.

```
$ rosrun soarwrapper soar2.py
```

- "Go" 명령으로 OP가 걸어가는 중에 'X' 가 그려진 종이를 OP의 얼굴 앞에 보여주면 영상 인식을 해서 제자리에 서고, 그 다음에 "Go" 명령에 대해서는 따르지 않는다.
- OP에게 'O' 가 그려진 종이를 얼굴 앞에 보여주면 불복종 모드가 해제되어 "Go" 명령을 내리면 앞으로 걸어간다. [그림 2-9]는 이러한 두 가지 상황을 나타낸다.



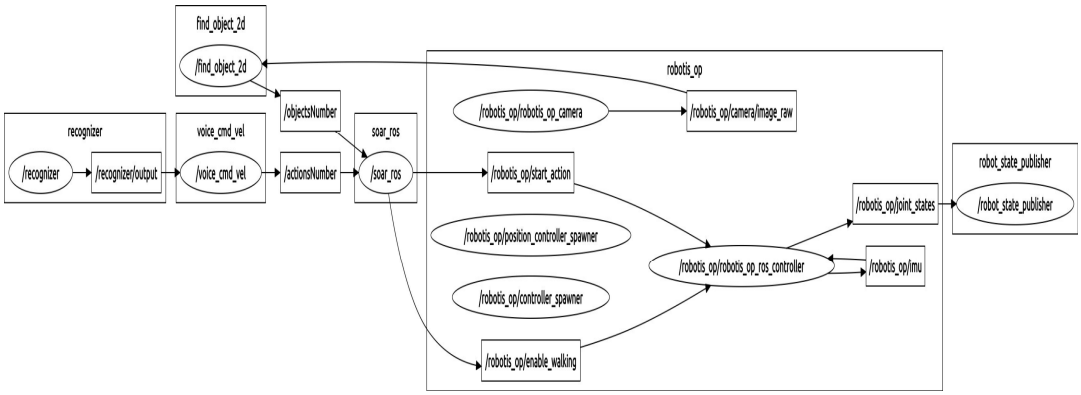
[그림 2-9] OP가 Soar-ROS 패키지를 이용하여 HRI 태스크를 실행하는 모습(왼쪽은 보행을 하다가 'X' 마크를 보고 제자리에 멈춰서는 순간을, 오른쪽은 'O' 마크를 보고 다시 보행을 하는 순간을 나타냄)

5. 전체 HRI 패키지의 그래프 구조를 보며 프로그램의 흐름을 확인한다.

- 전체 HRI 패키지의 노드 구조를 보기 위해 새로운 터미널을 열고 다음 명령을 실행한다.

```
$ rqt_graph
```

그러면 [그림 2-10]과 같이 패키지의 전체 노드 구조도를 확인하고 분석해 볼 수 있다.



출처: rqt_graph. 스크린샷.
[그림 2-10] OP에서 구현된 Soar-ROS 패키지의 노드 구조도