

로봇 소프트웨어 개발 도구 – (1) ERSP

■ 박광현

(광운대학교 정보제어공학과)

1. 서 론

Evolution Robotics사에서 개발한 ERSP (Evolution Robotics Software Platform)[1]는 로봇 소프트웨어를 개발하기 위한 통합 개발환경이라고 하기에는 무리가 있지만, 하드웨어와 응용 프로그램 사이의 다양한 컴포넌트를 제공함으로써 로봇의 구동부터 지능형 서비스에 이르기까지 로봇의 여러 가지 기능을 손쉽고 빠르게 개발할 수 있는 툴킷이다.

ERSP의 유연한 소프트웨어 구조는 다양한 드라이버와 행위(Behavior), 작업(Task)과 같은 소프트웨어 모듈을 포함하고 있어 로봇의 하드웨어 사양과 원하는 동작을 설정하면 해당 동작을 위한 함수들이 설정한 하드웨어에 맞게 동작한다.

본 원고에서는 로봇 소프트웨어 개발자가 개발도구를 선택하는 데 도움을 주기 위하여 ERSP의 구성과 특징, 성능 평가 [2]를 분석하여 소개하고자 한다.

2. ERSP의 구성(3)

ERSP는 그림 1과 같이 비전(Vision), 네비게이션(Navigation), 인터랙션(Interaction), 아키텍처(Architecture) 등 4개의 독립된 모듈이 하나의 패키지로 구성된 C++ 라이브러리이다. ERSP는 로봇의 행동을 제어하기 위한 강력한 아키텍처를 제공하는데, 아키텍처는 세 개의 계층으로 구성되어 있고 한 개 이상

의 계층을 사용하여 효과적으로 알고리즘을 구현할 수 있다. 비전(Vision)은 물체 인식 기능을 제공하고, 네비게이션(Navigation)은 새로운 환경에 대한 지도를 작성하고 로봇의 위치를 추정하며 경로를 생성하여 로봇을 다른 장소로 이동시키는데 사용한다. 인터랙션(Interaction)은 사람과 로봇의 상호 작용을 가능하게 하고, 감정 모델을 통해 로봇의 감정을 표현할 수 있다.

2.1 아키텍처(Architecture)

ERSP의 아키텍처는 크게 3개의 계층으로 구성되어 있다. HAL(Hardware Abstraction Layer)은 하드웨어 디바이스와 OS에 대한 의존도를 제거하는 인터페이스를 제공한다. 로봇의

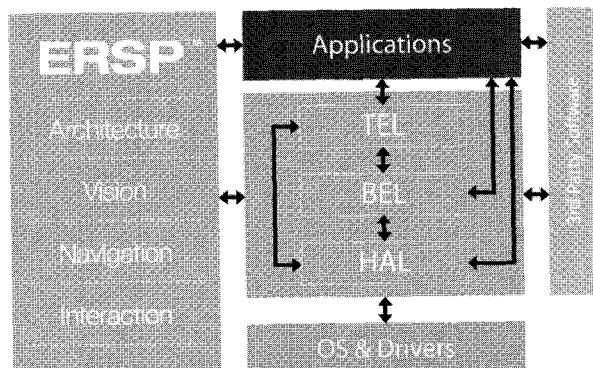


그림 1. ERSP의 전체 구조

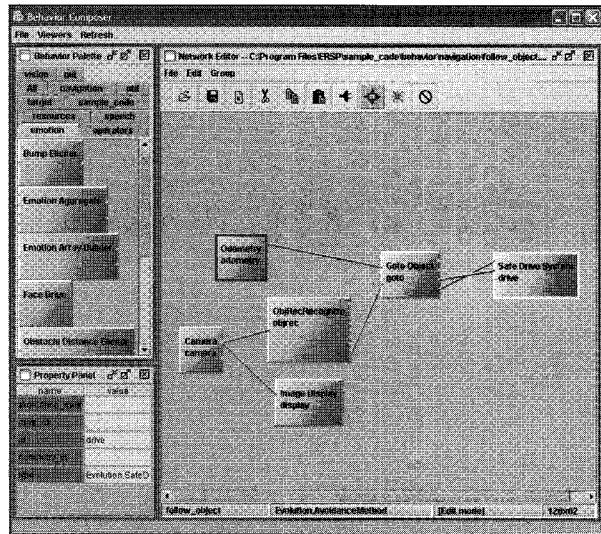


그림 2. Behavior Composer

하드웨어 사양과 구조를 XML 파일에 정의하고, 자원 관리자(Resource Manager)가 이를 참조하여 하드웨어를 구동시킨다. 이 파일이 잘못되어 있으면 로봇 제어 모듈이 동작하지 않고, 기술된 내용에 따라 로봇의 실행에 많은 영향을 미치기 때문에 정확한 내용이 필요하다. 또한, 각 장치에 대한 사양은 XML 언어로 기술된 스키마 파일에 기록되어 있다. 이 파일에는 I/O의 타입, 입출력 설정, 기본 값 등의 정보가 기술되어 있고 I/O 드라이버의 경로를 설정하도록 되어 있다.

BEL(Behavior Execution Layer)은 센싱, 의사결정, 자율 행동을 위해 사용되는 모듈과 하부 구조를 포함하고 있다. 행위(Behavior)는 여러 가지 작업(Task)과 응용을 개발하는 기본적인 구성 컴포넌트이며, 그림 2와 같은 Behavior Composer를 통해 다양한 행위를 조합할 수 있다.

TEL(Task Execution Layer)은 목적지향의 복잡한 작업을 개발하기 위한 하부 구조를 제공한다. TEL은 자극-반응 관계로 이루어진 행위(Behavior)나 세어 루프에 대해서는 비효율적이지만, 연속적인 일련의 행동에 효율적이다. TEL은 C++ 혹은 Python 인터페이스를 제공하며, BEL 계층을 거치지 않고도 HAL 계층을 바로 사용할 수 있다.

2.2 비전(Vision)

ERSP에서 제공하는 ViPR(Visual Pattern Recognition)은 하나의 저가 카메라를 이용하여 물체를 인식하기 위해 만들어진 모듈로서 실제 환경에서강인하고 정밀한 인식이 가능하다. SIFT 특징 점을 기본으로 한 인식 알고리즘을 사용하기 때문

에 특징 점이 있는 물체에 대해서는 높은 인식률을 보이지만, 특징이 없는 물체는 잘 인식되지 않는다. 이 외에도 사람이나 물체의 움직임을 식별하는 움직임 흐름(Motion Flow)과 색상 분할(Color Segmentation) 기능도 같이 제공된다.

2.3 네비게이션(Navigation)

ERSP는 저가의 센서와 하드웨어를 사용하여 자율 주행할 수 있는 vSLAM(Visual Simultaneous Localization And Mapping) 모듈을 제공한다. 영상 정보를 이용하여 지도를 작성하고 현재의 자기 위치를 인식할 수 있는데, 저렴한 가격에 비해 높은 정확성을 가지고 있다. 또한, 작성된 지도에서 주어진 위치로 경로를 생성하는 기능을 제공하는데, 장애물을 피해갈 수 있는 최단 경로를 생성하거나 랜드마크를 이용하여 이전에 갔던 경로를 안전하게 갈 수도 있다. 초음파 센서나 IR 센서를 이용하여 장애물 및 위험 지역을 회피하는 기능도 있으며, 물체 추종, 색상 추종, 움직임 추종과 같은 목표물 추종도 가능하다.

2.4 인터랙션(Interaction)

ERSP는 인간과 로봇 간의 상호작용을 위해 로봇의 내부 감정 상태를 표현하는 얼굴 애니메이션, 사람 검출과 추종, 간단한 제스처 인식 등이 가능하다.

3. 얼굴 인식 성능 평가

6명의 얼굴을 촬영한 동영상 파일을 프레임 별로 나누어 프레임 단위의 인식률, 동영상 파일 단위의 인식률, 프레임 당 평균 인식 시간을 측정하였다. 또한, 얼굴 회전에 따른 인식률을 측정하기 위해 6명의 사람을 대상으로 2m 위치에서 x, y, z축 방향으로 각각 $-15^\circ, -10^\circ, -5^\circ, +5^\circ, +10^\circ, +15^\circ$ 회전한 경우의 인식률을 측정하였다.

학습 모델을 만드는 방법은 두 가지를 사용하였다. 첫 번째는 ERSP에서 제공하는 방법대로 6명의 학습 영상을 모두 사용해 하나의 모델 파일을 만드는 방법이다. 두 번째는 각 사람 별로 모델 파일을 따로 만들고, 인식 할 때 각 모델이 제공하는 유사도 값을 비교하여 유사도가 가장 높은 모델을 인식 결과로 하는 방법이다.

3.1 정면 얼굴 인식

영상 데이터를 얻기 위해 각 사람 별로 1~3m 사이에서 자유롭게 움직인 동영상 5개씩과 1, 2, 3m 각각에서 약간씩 움직인

동영상을 각각 3개씩 얻었다. 즉, 사람 별로 총 4가지 종류의 동영상 14개씩을 촬영하였다. 모델을 구축하기 위해 각 사람 별로 동영상에서 추출한 하나의 사진을 사용하였으며 배경을 제거하고 얼굴만 남도록 하였다. 모델을 구축한 후에는 동영상의 나머지 프레임을 이용하여 얼굴 인식을 수행하였다. 프레임 단위의 인식률은 성공한 프레임의 개수를 전체 프레임의 개수로 나누어 구하였다. 또한, 각 프레임 별로 얼굴 인식을 하는데 걸리는 시간의 평균을 구하였는데, 영상을 읽어오는 등의 과정은 포함하지 않고 순수하게 인식에 걸린 시간만 측정하였다. 동영상 파일 단위의 인식률은 하나의 동영상에 대해 각 프레임 별 인식률의 결과가 사람 A라고 가장 많이 나왔을 경우 사람 A로 인식된 것으로 계산하였다.

(1) 1~3m 사이에서 움직인 경우

하나의 모델을 사용한 경우, 노트북 컴퓨터(CPU: 1.5Ghz, RAM: 1GByte)에서 프레임 당 평균 인식 시간은 135.3ms로 측정되었다. 사람에 따라 인식률의 차이가 크고, 하나의 모델을 사용한 경우와 사람 별 모델을 사용한 경우의 인식률에도 차이가 있음을 알 수 있다(표 1, 표 2).

표 1. 하나의 모델을 사용한 경우

사람	프레임 단위 인식률	동영상 단위 인식률
A	35.06%	100%
B	77.81%	100%
C	84.27%	100%
D	41.57%	100%
E	40.67%	100%
F	64.49%	100%
평균	56.61%	100%

표 2. 사람 별로 다른 모델을 사용한 경우

사람	프레임 단위 인식률	동영상 단위 인식률
A	90.66%	100%
B	52.23%	100%
C	45.12%	100%
D	33.79%	100%
E	37.12%	100%
F	44.68%	100%
평균	50.54%	100%

표 3. 하나의 모델에 대한 프레임 단위 인식률

사람	1m에서의 프레임 단위 인식률	2m에서의 프레임 단위 인식률	3m에서의 프레임 단위 인식률	프레임 단위 평균 인식률	프레임 당 평균 인식시간 (ms)
A	92.62%	82.55%	11.86%	62.34%	138.36
B	100.00%	96.87%	8.50%	68.46%	132.39
C	99.78%	56.38%	8.72%	54.96%	133.35
D	100.00%	58.39%	9.62%	56.00%	121.44
E	98.66%	73.83%	1.12%	57.87%	125.59
F	87.02%	63.09%	2.24%	50.78%	152.09
평균	96.35%	71.85%	7.01%	58.40%	133.87

표 4. 사람 별 모델에 대한 프레임 단위 인식률

사람	1m에서의 프레임 단위 인식률	2m에서의 프레임 단위 인식률	3m에서의 프레임 단위 인식률	프레임 단위 평균 인식률	프레임 당 평균 인식시간 (ms)
A	83.10%	91.33%	99.57%	91.33%	1028
B	8.88%	83.77%	16.22%	36.29%	959
C	18.67%	61.33%	14.68%	31.56%	989
D	45.77%	58.47%	10.90%	38.38%	723
E	62.00%	78.23%	2.23%	47.49%	783
F	56.00%	69.10%	6.67%	43.92%	1360
평균	45.74%	73.71%	25.04%	48.16%	974

표 5. 하나의 모델에 대한 동영상 단위 인식률

사람	1m에서의 동영상 단위 인식률	2m에서의 동영상 단위 인식률	3m에서의 동영상 단위 인식률	동영상 단위 평균 인식률
A	100.00%	100.00%	100.00%	100.00%
B	100.00%	100.00%	100.00%	100.00%
C	100.00%	100.00%	67.00%	88.90%
D	100.00%	100.00%	67.00%	88.90%
E	100.00%	100.00%	33.00%	77.80%
F	100.00%	100.00%	67.00%	88.90%
평균	100.00%	100.00%	72.20%	90.70%

표 6. 사람 별 모델에 대한 동영상 단위 인식률

사람	1m에서의 동영상 단위 인식률	2m에서의 동영상 단위 인식률	3m에서의 동영상 단위 인식률	동영상 단위 평균 인식률
A	100.00%	100.00%	100.00%	100.00%
B	100.00%	100.00%	100.00%	100.00%
C	100.00%	100.00%	100.00%	100.00%
D	100.00%	100.00%	67.00%	88.90%
E	100.00%	100.00%	67.00%	88.90%
F	100.00%	100.00%	100.00%	100.00%
평균	100.00%	100.00%	88.90%	96.30%

표 7. 얼굴 회전에 따른 인식률

	하나의 모델에 대한 평균 인식률	사람 별 모델에 대한 평균 인식률
x-5°	100%	98%
x-10°	100%	93%
x-15°	91.67%	90%
x+5°	93.33%	92%
x+10°	90%	90%
x+15°	85%	73%
y-5°	96.67%	93%
y-10°	93.33%	83%
y-15°	78.33%	80%
y+5°	98.33%	98%
y+10°	98.33%	85%
y+15°	76.67%	75%
z-5°	96.67%	95%
z-10°	91.67%	85%
z-15°	86.67%	63%
z+5°	83.33%	82%
z+10°	85%	83%
z+15°	68.33%	45%

(2) 1, 2, 3m 각각에서 약간씩 움직인 경우

하나의 모델을 사용한 경우에는 거리가 멀어질수록 인식률이 낮아짐을 알 수 있는데, 3m의 경우 얼굴이 매우 작아서 ERSP에서 요구하는 특정 점의 개수를 얻기에 부족한 것으로 판단된다. 사람 별로 다른 모델을 사용한 경우에는 1m 거리에

서의 인식률보다 2m 거리에서의 인식률이 높게 나와서 하나의 모델을 사용하였을 때와는 다른 경향을 보인다(표 3~6).

3.2 얼굴 회전에 따른 인식률

카메라에서 2m 떨어진 거리에서 정면 얼굴 영상을 획득하여 모델을 만들었다. 테스트 영상은 각 사람 별로 2m 거리에서 x, y, z 축 방향으로 각각 -15°, -10°, -5°, +5°, +10°, +15° 회전한 경우에 대해 각각 얼굴 영상 10장씩을 획득하였다.

기울어진 각도가 클수록 인식률이 낮아지는 것을 확인할 수 있는데, x축의 경우에는 회전을 하더라도 정면이기 때문에 인식률이 높고, y축과 z축의 경우에는 각도에 따라 인식률의 차이를 보였다(표 7).

4. 네비게이션 성능 평가

ERSP를 이용하여 실내에서 지도를 작성하고, 만들어진 지도의 정확도를 측정하였다. 지도 상의 원하는 지점으로 보내는 실험을 하여 성공률을 측정하였으며, 임의의 장애물을 설정하고 장애물 회피 능력을 측정하였다. 또한, 로봇을 임의의 위치로 옮겼을 때 자신의 위치를 파악하는 능력을 측정하였다. 실험에는 파이오니어 P3-DX-SH 로봇과 P3-AT 로봇을 사용하였다.

4.1 지도의 정확도 측정

복잡한 환경에서 지도를 작성할 경우 실제 환경의 크기를 측정하는데 어려움이 있기 때문에 생성된 지도의 정확도를 평가하기가 어렵고, 랜드마크를 생성하는 개수에 따라 정확도가 달라지게 된다. 따라서 vSLAM에 의해 생성된 지도의 정확도를 측정하기 위해 가능한 간단한 환경에서 일정 시간 동안 지도를 작성하고, 생성된 랜드마크의 개수를 측정하여 생성된 지도와 실제 환경 간의 오차를 측정하는 과정을 지도 작성 시간을 다르게 하면서 여러 번 측정하였다.

(1) 넓이를 기준으로 정확도 평가

ERSP에서 제공하는 네비게이션 툴을 사용하여 지도를 작성할 때 자동탐사를 사용하면 아주 많은 시간이 소요될 뿐더러 작성된 지도가 실제와 큰 차이가 있었다. 또한, 탐사 시간을 600초로 설정해도 실제 지도 작성 시간은 이보다 훨씬 오래 걸리는 것을 볼 수 있었다. 따라서 지도 작성은 탐사를 사용하지 않고 조이스틱으로 직접 가장자리 부분 위주로 로봇을 움직

여 생성하였다. 지도를 생성하였을 때, 로봇이 탐지한 영역은 실제 환경보다 더 넓게 측정되었는데, 이는 로봇을 원하는 지점(가장자리)으로 보냈을 때 벽이나 물체에 부딪치는 원인으로 판단된다. 생성된 지도와 실제 환경 간의 오차는 약 0.423%로 측정되었다.

(2) 랜드마크의 정확도를 기준으로 평가

생성된 지도에서 랜드마크의 정확성을 기준으로 평가하면 지도의 정확도는 75~76%이다.

4.2 임의의 특징 점으로 로봇을 보냈을 경우의 성공률

생성된 지도에서 임의의 특징 점 30곳으로 로봇을 보내고 해당 위치로 가는지를 평가하였다. 두 가지 로봇 각각에 대해 총 30회와 60회의 실험을 하였는데, 성공률은 각각 73%와 78.3%로 측정되었다.

실패한 경우는 랜드마크의 위치가 부정확하여 로봇이 해당 목표지점에 도착하지 못하고 잘못된 방향으로 이동하며 장애물에 충돌하여 더 이상의 네비게이션이 불가능한 경우였다.

4.3 장애물 회피 성공률 측정

로봇을 임의의 위치로 이동시킬 때 지도에 없는 장애물이 나타날 수 있다. 이 때, 로봇이 그 장애물을 검출하여 피해갈 수 있는지를 실험하였다. 총 30회의 실험 중에서 장애물 회피는 18번 성공하였다.

4.4 키드냅(Kidnap)된 로봇이 자기 위치를 파악하는데 걸리는 시간 측정

vSLAM을 이용한 ERSP는 로봇이 키드냅된 경우 자신의 위치를 파악할 수 있는 알고리즘을 제공한다. 키드냅된 로봇이 자신의 위치를 찾는 방법으로 기존에 저장된 랜드마크들과 새로 얻은 시각적 이미지를 비교하는 방법을 사용하기 때문에 랜드마크가 많으면 많을수록 위치를 파악하는 시간이 줄어들게 된다.

두 가지 로봇에 대해 각각 총 30회 실험을 하였는데, 성공률은 각각 100%와 80%로 측정되었고, 성공하였을 때 평균 소요 시간은 각각 28.58초와 38초로 측정되었다.

4.5 자원 점유율

1.2GHz CPU와 1GB의 RAM을 가진 노트북에서 지도 작성 시의 CPU 점유율은 80~100%, RAM 점유율은 40~50%로 측정

되었고, 기타 작업 시의 CPU 점유율은 80~100%, RAM 점유율은 12~25%로 측정되었다.

5. 결 론

5.1 ERSP의 장점

- * 소니 AIBO, Bandai, ZMP, SKTelecom, 유진 로봇 등에 적용 한 사례를 가지고 있다.
- * 윈도우 및 리눅스를 지원하며, 체계적인 소프트웨어 구조로 높은 유연성과 이식성, 재사용성을 가진다.
- * 하드웨어와 로봇의 행위에 대한 인터페이스가 잘 설계되어 있어 로봇 응용을 쉽게 개발할 수 있다.
- * 제공하는 고성능의 알고리즘을 쉽게 사용할 수 있다.
- * 강력한 물체 인식 알고리즘을 제공하며, 학습 영상을 하나만 사용하더라도 거리, 기울기, 가림에 강인한 인식을 할 수 있다.
- * 비교적 간단한 저가의 장비(카메라 1개, 엔코더)로 복잡한 공간에 대한 지도를 작성할 수 있다.
- * 키드냅 시 랜드마크를 통해 비교적 빠른 시간 내에 자신의 위치를 파악한다.

5.2 ERSP의 단점

- * Visual IDE기반의 개발 툴(예: eclipse)과 연동이 어렵다.
- * ERSP가 설치되지 않은 환경에서는 프로그램이 동작하지 않는다.
- * 새로운 컴포넌트를 개발하기가 쉽지 않으며, 숙련된 개발자가 필요하다.
- * 시뮬레이터 등과 같은 체계적인 개발도구가 지원되지 않아 실제 하드웨어가 없으면 사용할 수 없다.
- * 형태가 변하지 않는 물체는 잘 인식하지만, 형태가 약간씩 변할 수 있는 경우(예: 얼굴 인식)에는 인식률이 조금 떨어지는 경향이 있다.
- * 장애물이 있는 곳에서 지나치게 오랫동안 머무르고 다른 방향을 향해 탐색하지 않거나, 이미 탐색한 공간을 계속 반복적으로 탐색하는 등 탐사를 위한 전략이 지능적이지 않다.
- * 저가용 센서를 위해 개발되었다고는 하지만 정확성을 추구하는 응용을 위해 더욱 정밀한 센서를 위한 라이브러리를 제공할 필요가 있다.
- * SIFT를 사용하기 때문에 시각적 특징 점이 없는 곳에서는 지도가 작성되지 않는다. 즉, 주위 환경이 비슷한 경우(예:

병원 복도)에는 적용하기가 어렵다.

본 원고는 정보통신부 정보통신연구개발사업의 연구결과를 바탕으로 작성되었음을 밝힙니다. 실험을 도와 주신 카이스트의 도준형씨, 전문진씨, 김성민씨, 최현종씨, 한양대학교의 류광근씨, 박영빈씨께 감사 드립니다.

참고문헌

- [1] <http://www.evolution.com>
- [2] 정보통신부, ERSP 분석 결과보고서, 2007년 7월
- [3] ERSP 사용자 매뉴얼

● 저자 약력



박광현

- 1994년 2월 한국과학기술원 전자전산학과(학사)
- 1997년 2월 한국과학기술원 전자전산학과(석사)
- 2001년 2월 한국과학기술원 전자전산학과(박사)
- 2008년 3월~현재 광운대학교 정보제어공학과 조교수
- 관심분야 : 서비스 로봇, 보조공학 시스템, 인간-로봇 상호작용, 기계 학습, 패턴 인식