

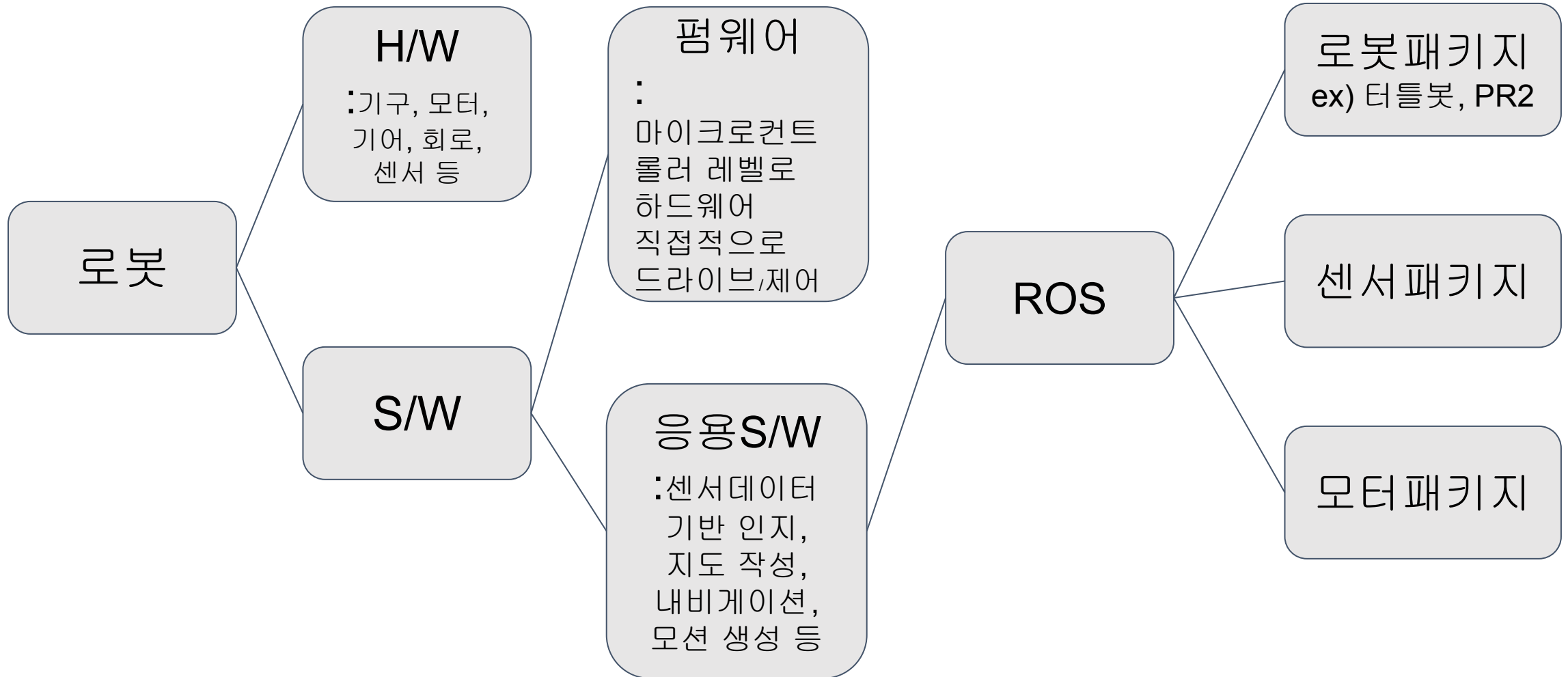
# ROS 로봇프로그래밍

-ROS1 완독하기 챌린지-

김예나

## Ch8. 로봇, 센서, 모터

## 8.1 로봇 패키지



## 8.1 로봇 패키지

### ROS 활용 로봇 종류

- 매니플레이터
- 모바일 로봇
- 자동 주행 자동차
- 휴머노이드
- 무인 항공기(UAV)
- 무인 잠수함(UUV)
- 무인 표면 주행차(UWV)

### 로봇 패키지 구성

= 로봇 구동 드라이브 노드  
+ 장착된 센서 데이터의 취득 및 활용 노드  
+ 원격 조정 노드  
(+역기구학 노드 +내비게이션 노드 등)

<사용하려는 로봇 패키지가 ROS 공식 패키지일 경우>

패키지 찾기

- ⇒ <http://robots.ros.org/> 에서 확인
- ⇒ 명령어 `$apt-cache search ros-noetic`
  - 전체 **ros**패키지 리스트에서 찾기 가능
- ⇒ 리눅스 **GUI** 패키지 매니저 프로그램 **synaptic** 구동
  - 'ros-noetic' 단어로 검색

패키지 설치 예시

- ⇒ `$sudo apt-get install (ros-noetic-pr2-desktop)`  
`$sudo apt-get install (ros-noetic-turtlebot3`  
`ros-noetic-turtlebot3-msgs ros-noetic-turtlebot3-simulations)`

<사용하려는 로봇 패키지가 ROS 공식 패키지 아닐 경우>

- `$cd ~/catkin_ws/src` → 캐킨 빌드 시스템의 사용자 소스 폴더로 이동
- `$ hg clone http://code.google.com/p/amor-ros-pkg/`  
→ 리포지토리로부터 내려받기

## 8.2 센서 패키지

**센서가 중요한 이유?** 주변 환경을 인식하고 의미 있는 정보만을 추출하여 계획 및 사고 판단 할 수 있을 때 비로소 로봇

### 센서의 종류

#### 거리센서

- 레이저 기반의 거리 센서 및 적외선 거리 센서  
: LDS(Laser Distance Sensor),  
LiDAR(Light Detection And Ranging),  
LRF(Laser Range Finders)

- 3차원 거리 센서: 리얼센스, 키넥트, 액션 등  
컬러 카메라: 사용자나 물체 인식 등

관성 센서: 위치 추정

마이크로폰: 음성 인식

토크 센서: 토크 제어

기타 등등

### ROS

⇒ 센서 드라이버 및 라이브러리 사용 가능 개발환경 제공

LDS, 3D센서, 카메라 등 데이터 양 ↑, 상당한 사양 요구

⇒ 마이크로프로세서로 무리

⇒ PC 이용

⇒ 드라이버, OpenNI, OpenCV 등 포인트 클라우드 처리, 영상 처리 등에 필요한 라이브러리 필요!

## 8.2 센서 패키지

### 센서 패키지의 분류

- 1D Range Finders: 저가의 로봇을 만들 때 사용할만한 적외선 방식의 직선거리 센서들
- 2D Range Finders: 2차 평면상의 거리를 계측할 수 있는 센서로 주로 내비게이션에 많이 사용
- 3D Sensor: intel사의 RealSense, Microsoft사의 Kinect, ASUS사의 Xtion 과 같은 3차원 거리 측정에 사용되는 센서들
- Audio/Speech Recognition: 현재 음성인식 관련 부분은 매우 적지만, 지속해서 추가될 것으로 보임
- Cameras: 물체인식, 얼굴인식, 문자판독 등에 많이 사용되는 카메라의 드라이버, 각종 응용 패키지를 모아 둠
- Sensor Interfaces: USB 및 웹 프로토콜을 지원하는 센서는 매우 적음. 아직까지도 많은 센서들은 마이크로프로세서에서 정보를 쉽게 얻을 수 있는 센서가 많음. 이러한 센서는 마이크로프로세서의 UART 및 미니 PC 계열에서 ROS와의 연결을 지원함.
- 그 외: ROS 센서 위키 페이지에 여러 센서 패키지가 공개되어 있음

## 8.3 카메라

**카메라 = 로봇의 눈**

**활용** : 카메라의 이미지를 이용한 물체 인식, 얼굴 인식, 카메라 두 대(스테레오 카메라)를 이용하여 서로 다른 두 이미지 차이에서 얻는 거릿값, 이를 응용한 3차원 지도 생성인 비주얼 슬램, 단안 카메라 비주얼슬램, 색상 인식, 물체 추적 등

### 8.3.1 USB(UVC) 카메라

– 패키지 종류

- libuvc-camera: UVC 표준을 사용하는 카메라들을 사용하기 위한 인터페이스 패키지
- uvc-camera: 비교적 상세한 카메라 설정 변경 있어 매우 편리, 스테레오 카메라에 적절
- usb-cam: Bosch에서 사용하는 매우 간단한 카메라 드라이버
- freenect-camera, openni-camera, openni2-camera: Kinect나 Xtion 같은 심도 카메라를 위한 패키지, 컬러 카메라 포함 RGB-D카메라라고도 불림
- camera1394: IEEE 1394 규격인 FireWire를 이용하는 카메라를 위한 드라이버
- prosilica-camera: 연구용으로 많이 사용되는 AVT 사의 prosilica 카메라에 사용
- pointgrey-camera-driver: Point Grey 카메라를 위한 드라이버
- camera-calibration: OpenCV의 캘리브레이션 기능을 응용한 카메라 캘리브레이션 관련 패키지

## 8.3 카메라

### 8.3.1 USB(UVC) 카메라

#### – USB 카메라 테스트

- 1) USB카메라 포트에 연결
- 2) lsusb 명령어 : 카메라 접속 확인
- 3) uvc camera 패키지 설치 : `$sudo apt-get install ros-noetic-uvc-camera`
- 4) image 관련 패키지 설치 : `$sudo apt-get install ros-noetic-image-*`  
`$sudo apt-get install ros-noetic-rqt-image-view`
- 5) uvc\_camera 노드 실행 : `$ roscore $roslaunch uvc_camera uvc_camera_node` (캘리브레이션 관련 경고 일단 무시)
- 6) 토픽 메시지 확인 : `rostopic list` → /camera\_info 카메라 정보 , /image\_raw 이미지 정보 퍼블리시 되고 있음

#### – 이미지 정보 확인

- 1) image\_view 노드로 확인 : `$roslaunch image_view image:=/image_raw`
- 2) rqt\_image\_view 노드로 확인 : `$rqt_image_view image:=/image_raw`
- 3) RViz로 확인 : `$rviz` [Add] – [By display type] – [image] – [image topic]: '/image\_raw' 변경



## 8.3 카메라

### 8.3.1 USB(UVC) 카메라

- 원격 이미지 전송 : 로봇에 장착된 카메라의 이미지 정보를 원격지의 다른 컴퓨터에서 확인하는 방법
  - 원격 탐사 로봇, 화상회의 로봇, 영상을 실시간으로 네트워크에 전송할 수 있는 웹캠 등 활용

카메라가 연결된 컴퓨터 : ROS마스터

\$ gedit ~/.bashrc → ROS\_MASTER\_URI, ROS\_HOSTNAME 변수 수정

ex)export ROS\_MASTER\_URI = <http://192.168.1.100:11311> , export ROS\_HOSTNAME = [192.168.1.100](http://192.168.1.100)

\$ roscore

\$ roslaunch uvc\_camera uvc\_camera\_node

원격 컴퓨터 :

\$ gedit ~/.bashrc 변수 수정 → ROS\_MASTER\_URI : 카메라 연결된 컴퓨터의 IP

ROS\_HOSTNAME : 원격 컴퓨터의 IP

ex)export ROS\_MASTER\_URI = <http://192.168.1.100:11311> , export ROS\_HOSTNAME = [192.168.1.120](http://192.168.1.120)

\$ roslaunch image\_view image\_view image :=/image\_raw

## 8.3 카메라

### 8.3.1 USB(UVC) 카메라

- 카메라 캘리브레이션 : 스테레오 카메라나 이미지를 가지고 거릿값 측정 및 물체인식 등의 영상 처리시, 카메라마다 다른 렌즈 특성, 렌즈와 이미지 센서와 거리, 뒤틀린 각도 등을 고려한 카메라 고유 보정 파라미터 찾는 것!

카메라 캘리브레이션 패키지 설치

```
$ sudo apt-get install ros-noetic-camera-calibration
```

```
$ rosrn uvc_camera uvc_camera_node
```

체스보드 준비

[http://wiki.ros.org/camera\\_calibration/Tutorials/MonocularCalibration?action=AttachFile&do=view&target=check-108.pdf](http://wiki.ros.org/camera_calibration/Tutorials/MonocularCalibration?action=AttachFile&do=view&target=check-108.pdf)

```
$ rosrn camera_calibration cameracalibrator.py -size 8x6 -square (네모 한칸 크기|mm)0.024 image:=/image_raw  
camera:=/camera
```

카메라 기준 체스보드를 앞/뒤로 움직이기, 비스듬히 기울이기 등 수행 - CALIBRATE버튼 활성화 - SAVE

카메라 파라미터파일 생성

```
$ cd /tmp      $ tar -xvzf calibrationdata.tar.gz // 압축풀기
```

```
$ mv ost.txt ost.ini // ost.txt파일을 ost.ini 파일로 이름 변경
```

```
$ rosrn camera_calibration_parsers convert ost.ini camera.yaml // 카메라 파마미터 파일(camera.yaml)생성
```

```
$ mkdir ~/.ros/camera_info
```

```
$ mv camera.yaml ~/.ros/camera_info/
```

## 8.4 심도 카메라(Depth Camera)

### -Depth Camera의 종류

- 1) ToF(Time of Flight) : 적외선을 방사하여 되돌아온 시간으로 거리 측정, 일반적으로 IR 송신부와 수신부가 쌍으로 있어서 각 픽셀에서 측정한 거리를 읽어내는 방식 → 구조광보다 비쌈
  - 피나소닉의 D-IMager, MESA Imaging사의 SwissRanger, Fotonics사의 FOTONIC-B70, pmdtechnologies사의 CamCube, Camboard, SoftKinect사의 DepthSense DS시리즈, Microsoft사의 Kinect2 등
- 2) 구조광(Structured Light) : 일관적 방사 패턴 방식 사용, ToF의 비싼 가격과 외부 간섭 등의 문제 해결 but 생산 중단 / 판매 중지 등의 분위기...
  - Microsoft 사의 Kinect, ASUS사의 Xtion, PrimeSense사의 Carmine, Capri 등, Occipital사의 Structure Sensor → PrimeSense사의 PrimeSense System on Chip(SoC) 사용
- 3) 스테레오(Stereo) : 사람의 좌우 눈처럼 양안시차를 이용하여 거리 측정, 일정 간격을 두고 두 개의 이미지 센서를 장착하고 있고, 두 장의 이미지 영상 차이를 이용해 거릿값 계산
  - Point Grey사의 Bumblebee, WithRobot사의 OjOcamStereo : 패시브 스테레오 카메라
  - 두 개의 적외선(IR) 이미지 센서와 적외선을 주사하는 프로젝터를 내장한 형태로 눈에 보이지 않는 적외선을 일정 패턴으로 주사하여 두 이미지 센서로 이를 받아 삼각 측량법을 통해 거리 측정 : 액티브 스테레오 카메라
    - Intel의 RealSense

## 8.4 심도 카메라(Depth Camera)

### -Depth Camera 테스트(Real Sense R200기준)

#### 1) RealSense 관련 패키지 설치

```
$sudo apt-get install ros-noetic-librealsense ros-noetic-realsense-camera
```

#### 2) r200\_nodelet\_default 런치 파일 실행

```
$roscore
```

```
$roslaunch realsense_camera r200_nodelet_default.launch
```

패키지 설치 안되거나 동작 안할 시, 리눅스 커널마다 달리 설정 필요→ <http://wiki.ros.org/librealsense> 참조

#### 3) Point Cloud Data 의 시각화

- 대상체와의 거리를 한 점으로 표시하고 이들 점들의 집합체가 구름 같다하여 point cloud data라 부름

- RViz 실행 - [Global Option]-[Fixed Frame]을 camera\_depth\_frame으로 변경 - [Add] - [PointCloud2]추가 - Topic으로 camera/depth/points 설정 및 크기 색상 지정 - 색상 기준 축기준 멀어질수록 보라색

#### - 관련 라이브러리

Point Cloud Library - point cloud를 사용하기 위한 API의 집합체, 필터링, 분할, 표면 재구성, 피팅 및 특징 추출 등을 수행하는 라이브러리 많이 사용됨

OpenNI - 인간과 기계의 의사소통이란 뜻으로, 드라이버 및 다양한 API 라이브러리, 인체골격 미들웨어 포함

## 8.5 레이저 거리 센서(LDS)

레이저 거리 센서(LDS)=광선 레이더(LIDAR)=레이저 레인지 파인더(LRF)=레이저 스캐너

레이저 광원을 이용하여 물체와의 거리를 측정하는데 사용하는 센서로, 고성능, 고속, 실시간 데이터 취득 가능, 로봇 분야에 많이 사용되는 센서로 SLAM, 물체나 사람 인식, 실시간성이 좋아 무인 자동차 등에 많이 사용

-Hokuyo의 URG 시리즈, 실외용 SICK 제품, 레이저 센서 여러 개 탑재한 Velodyne의 HDL 시리즈 등

-가격 비쌈 but 중국 제품(RPLIDAR 등) 40만원 안팎, 국내(HLS-LFCD2) 10만원대로 시장에 나옴

**-거리 측정 원리** : 레이저 광원이 물체에 반사되었을 때 발생하는 파장을 이용, 대부분 제어 및 가격 문제로 레이저 광원을 하나만 사용하기 때문에 이를 극복하기 위해 레이저 1개, 반사거울 1개, 모터로 구성됨. 비스듬히 기울어져 있는 거울을 회전시켜가며 레이저가 반사되어 돌아오는 시간을 측정

※주의할 점

- 1) 강한 레이저 빔 때문에 눈 손상될 수 있음, 클래스 숫자가 높아질수록 위험
- 2) 레이저 광원이 반사되어 돌아오는 것을 측정하기 때문에 반사되지 않으면 무소용 - 유리, 페트병, 유리컵 등
- 3) 수평면을 주사하기 때문에 센서 기준 수평면 위에 있는 물체만 검출 - 2D데이터임!

## 8.5 레이저 거리 센서(LDS)

### -LDS 테스트

#### 1) hls\_lfcd\_lds\_driver 패키지 설치

```
$sudo apt-get install ros-noetic-hls-lfcd-lds-driver
```

#### 2) LDS 연결 및 사용 권한 변경

```
$ls -l /dev/ttyUSB* (USB연결 포트 확인)
```

```
$sudo chmod a+rw /dev/ttyUSB0 (예를 들어 ttyUSB0) // 권한 설정(crw-rw-r-- → crw-rw-rw- 로 변경됨)
```

#### 3) hlds\_laser 런치 파일 실행

```
$roscore
```

```
$roslaunch hls_lfcd_lds_driver hlds_laser.launch
```

#### 4) scan 데이터 확인 : hlds\_laser 노드를 실행하면 '/scan'토픽으로 LDS값이 전송됨

```
$ rostopic echo /scan → frame_id: laser, 측정각, 각 증가 값, 측정 거리 최소/최대값, 각 측정 각도별 거리값 배열
```

#### 5) LDS 거릿값의 시각화

\$rviz : 오른쪽 상단 view type을 TopDownOrtho로 설정(XY평면뷰) - 왼쪽 상단 [Global Options] - [FixedFrame]을 'laser'로 변경 - [Add] - [Axes]추가 - 세부설정(Length 및 Radius)변경 - [Add] - [LaserScan]추가 - Topic 및 Color변경

-LDS의 활용 - SLAM : 로봇에 LDS를 장착하고, 로봇을 중심으로 장애물을 인식하고, 자신의 위치를 추정하며 맵을 작성

- 인물 및 이동 물체 검출

## 8.6 모터 패키지

ROS 위키의 Motors 페이지 : ros에서 지원하고 있는 모터 및 서버 컨트롤러 모아둔 페이지

– PhidgetMotorControl HC, Roboteq AX2550 Motor Controller, ROBOTIS Dynamixel 지원하는 패키지 등

## 8.7 다이나믹셀

모터 간을 연결하여 서로 통신할 수 있는 데이지 체인 방식으로 연결하여 제어 가능

→ 배선 정리 간편, 감속기어, 제어기, 구동부, 통신부로 구성된 모듈로써 위치, 속도, 온도, 부하, 전압, 전류 등을 피드백 받을 수 있음 + 속도 제어와 토크 제어도 가능

로봇에 활용 방법

1)통신 변환 디바이스 U2D2를 경유하여 제어명령을 액추에이터에 전달하는 방법

2)OpenCR과 같은임베디드 보드를 통해 직접 제어하는 방법

개발환경: DynamixelSDK – 3대 OS(linux, windows, macOS)지원,

c/c++,c#,python,java,MATLAB,LabVIEW 등 프로그래밍 언어 지원, 아두이노 및 ROS 패키지로도 지원

대표 패키지: dynamixel\_motor, arbotix, dynamixel\_workbench

## 8.8 공개 패키지 사용법

5000여개의 공개패키지 중 자신에게 필요한 패키지 검색, 설치, 사용법(예: find\_object\_2d)

- 1) <http://www.ros.org/browse/list.php> 웹 페이지 상단 ros버전 클릭
- 2) search란에 키워드 입력 후, 적절한 패키지 클릭
- 3) 빌드시스템, 제작자, 오픈 소스 라이선스의 종류 등 정보 확인
- 4) 상단의 noetic 버튼, 프로젝트 web페이지의 링크, 패키지 리포지토리 주소, 패키지 사용법, 패키지 의존성 확인

### 5) 의존성 패키지 설치

`$rospack list` // 필요한 패키지 설치 유무 확인

`$rospack find ~~~` //설치된 경우 패키지 확인

`$sudo apt-get install ros-noetic-~~~` //안 깔린 의존성 패키지 설치

### 6) 패키지 설치 두가지 방법

①바이너리 설치: `$sudo apt-get install ros-noetic-find_object_2d` //깔고자하는 패키지

②소스 설치: `$cd ~/catkin_ws/src $git clone ~URI~ $cd ~/catkin_ws/ $catkin_make`

### 7) 패키지 실행

`$roscore $roslaunch uvc_camera uvc_camera_node $roslaunch find_object_2d find_object_2d`

`image:=image_raw` // 검출 대상 이미지 저장후, 실행된 gui프로그램에 파일 드래그 앤 드롭

`$rostopic echo /object` // object 토픽 확인

`$roslaunch find_object_2d print_objects_detected` //검색된 대상의 정보 확인



–The end–