

collabot_manual_KR_1030

[프로그램 실행]

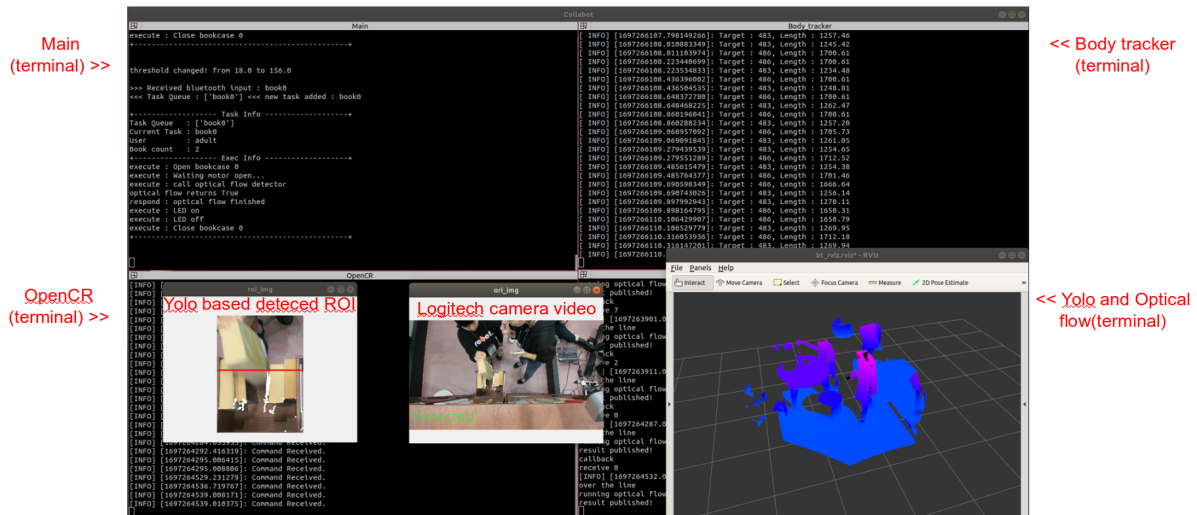
0. 선행 작업 : 카메라 두개 및 opencr 보드의 연결 확인 및 chkusb로 logitech 카메라 포트 확인(후술)

1. **roscore** : 터미널 열고, **roscore** 입력 후 엔터.
2. **runopt** : 터미널 열고, **runopt** 입력 후 엔터.
3. 모든 모듈이 준비되기 까지 약 5~6초 가량 소요됨.

[runopt 구성]

1. 4분할 터미네이터
 - a. 메인 스크립트(좌상단) : 앱으로 선택된 책을 출력 및 task queue에 저장, task queue의 각 타겟에 대하여 순차적으로 일련의 프로세스 진행과정 출력
책장 open → 대기 → detector 호출 → 응답 대기 → (구) 디버깅용 LED ON, OFF → 책장 close
 - b. OpenCR 모듈(좌하단) : 앱으로 책 선택시 “Command Received” 출력 및 데이터 main으로 전달, main에서 모터 open / close 수신 시 명령 수행
 - c. Body Tracking 모듈(우상단) : 실시간 Skeleton detection 정보 출력(바디 인덱스 및 신장 추정치)
 - d. Detection 모듈(우하단) : main에서 호출 시 logitech 카메라의 비디오에서 YOLO 기반 책장 ROI 검출, 해당 ROI 내에서 Optical Flow로 책을 꺼내갔는지 확인. 결과를 main에 반환
2. RVIZ
 - a. 실시간 Azure Kinect 카메라의 포인트클라우드 및 스켈레톤 데이터 출력.
3. rqt - dynamic reconfigure
 - a. Azure Kinect 포인트 클라우드 데이터의 rotation, translation을 수동조작
 - b. rotation, traslation preset 저장 및 불러오기
 - c. Collabot_main 탭에서 어른 아이 threshold 조절 가능.

4. OpenCV window (ori_img , roi_img) : Detection 모듈 시각화 용.



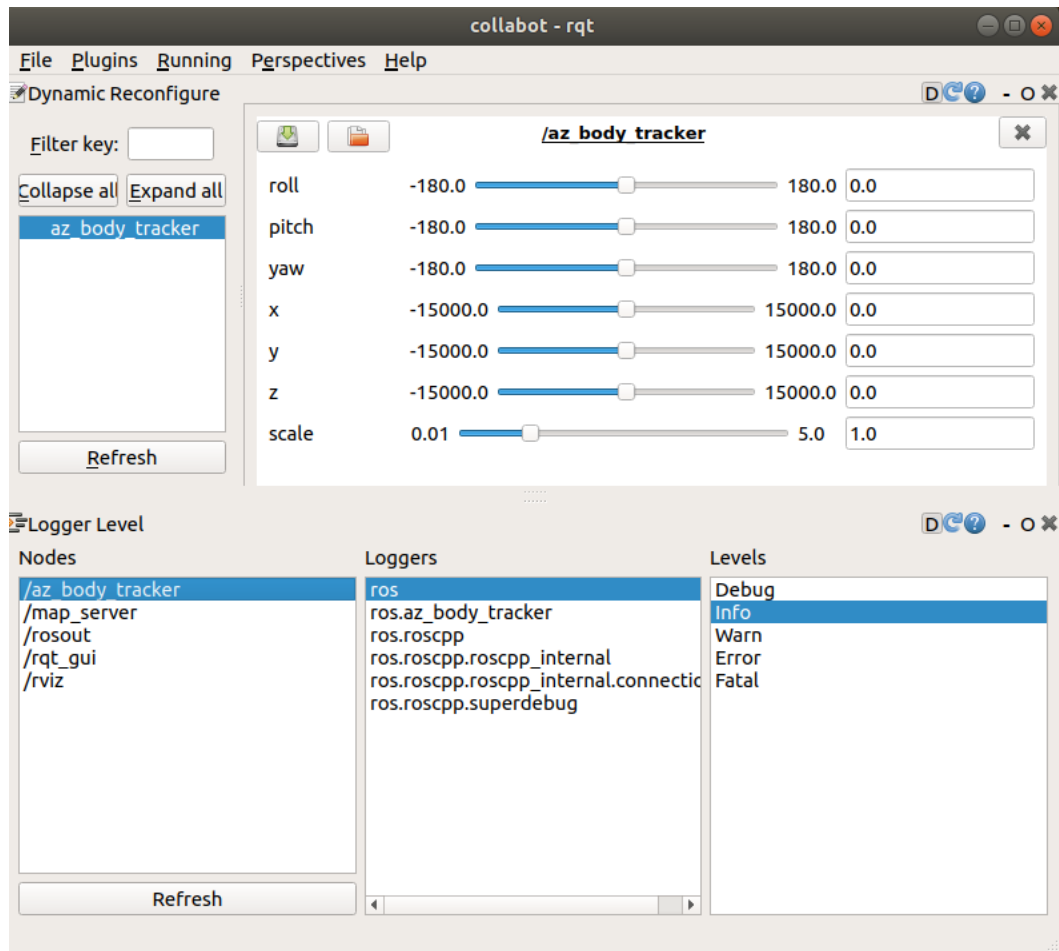
카메라 H/W pose setting

Azure Kinect - 은색

0. 카메라의 C 포트는 USB로 노트북과 연결. 카메라의 동그란 포트는 전원 어댑터에 연결.

* 아래부터는 최초 1회만 세팅 해주면 됨.

1. 터미널에서 **runopt** 또는 **roslaunch az_body_tracker run.launch** 명령어로 body tracker RVIZ 실행.
2. 포인트 클라우드 데이터에 가능한 책장이 안나오도록(FOV의 가려짐 유발)
3. 카메라 각도를 지평면으로부터 약 45~60도 사이의 각도로 맞춤.
4. 3번과 4번을 만족하면서 가장 사람 인식이 잘 되는 각도로 세팅.
5. 이후 rviz창과 함께 열리는 rqt dynamic reconfigure 창에서 roll pitch yaw 값을 조절하여 포인트클라우드의 바닥 면이 RVIZ ground plane과 매칭되도록 설정.



7. 좌상단 버튼 눌러서 preset 값 저장:

`~/catkin_ws1/src/az_body_tracker/params/preset.yaml`

Logitech - 검은색

로지텍 카메라는

1. 책장과 수평하면서
2. 지평면과 수직하면서
3. 책장 모서리보다 조금 앞으로 튀어나오게 하여 책장을 잘 인식하도록
4. 실행도중 인식이 잘 안되는 경우 각도의 조절이 필요할 수 있음.

카메라 S/W setting

1. 카메라 두 대를 노트북에 연결

2. 터미널 열고, **chkusb**로 logitech 카메라의 첫번째 포트 번호 확인 (아래의 경우 0번)

```
kist@kist-GL75-Leopard-10SDK:~$ chkusb
HD Webcam: HD Webcam (usb-0000:00:14.0-13):
    /dev/video2
    /dev/video3

Logitech BRIO (usb-0000:00:14.0-3.3):
    /dev/video0
    /dev/video1

Azure Kinect 4K Camera (usb-0000:00:14.0-3.4.1):
    /dev/video4
    /dev/video5
```

3. ~/yolov5 경로에 있는 detect_book_state.py 스크립트의 최하단 main에서 source 변수 값을 위에서 확인한 포트번호로 변경 후 저장.

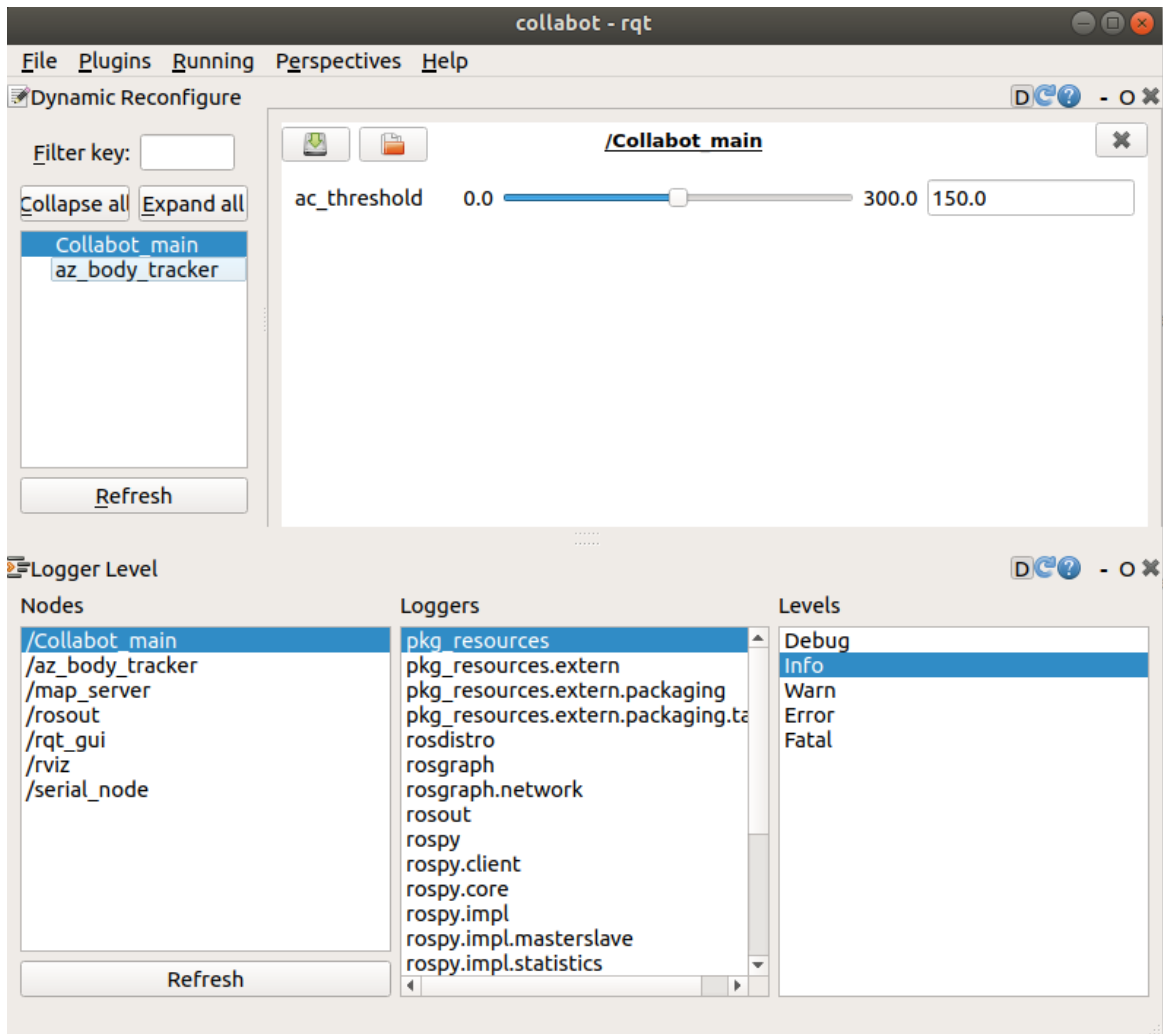
```
if __name__ == "__main__":
    rospy.init_node('detect_book_state')
    cv2.namedWindow('ori_img',cv2.WINDOW_NORMAL)
    cv2.namedWindow('roi_img',cv2.WINDOW_NORMAL)
    print("Starting...")
    # Using Camera
    source = 0
```

어른 아이 threshold

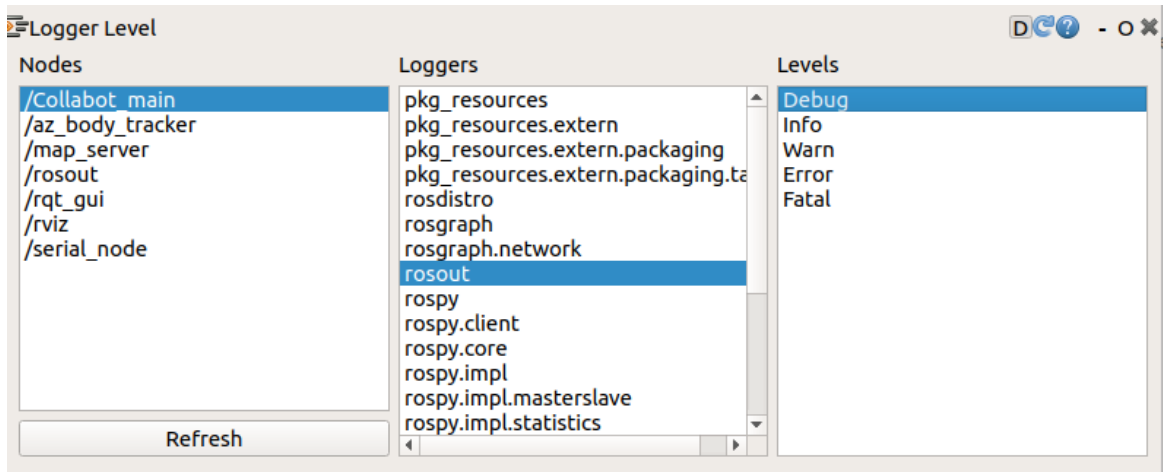
터미널에서 **runopt** 실행 후 프로그램이 완전히 실행 된 후 rqt 창에서 **refresh** 버튼 클릭.

목록에 **Collabot_main**을 클릭하면 ac_threshold 조절하는 트랙바가 나옴.

트랙바를 움직이거나, 우측의 박스에 원하는 값을 기입후 엔터.



어른아이 판정정보 디버깅은 logger level을 아래와 같이 설정. Levels의 기본값은 Info.



runopt 실행 구성

4분할 터미널

터미널 열고 **tconfig** 입력 후 엔터 >>> 통합 터미널에서 각 터미널에서 실행할 명령어가 정의되어 있음.

```
≡ config ×
home > kist > .config > terminator > ≡ config
1  [global_config]
2      suppress_multiple_term_dialog = True
3  [keybindings]
4  [layouts]
5      [[default]]
6          [[child1]]
7              parent = window0
8              profile = collabot
9              type = Terminal
10         [[window0]]
11             parent = ""
12             type = Window
13     [[collabot]]
14 >     [[child0]] ...
25 >     [[child1]] ...
31 >     [[child2]] ...
37 >     [[child5]] ...
43 >     [[terminal3]] ...
51 >     [[terminal4]] ...
59 >     [[terminal6]] ...
67 >     [[terminal7]] ...
75     [[collabot_opt]]
76 >     [[child0]] ...
87 >     [[child1]] ...
93 >     [[child2]] ...
99 >     [[child5]] ...
105 >     [[terminal3]] ...
113 >     [[terminal4]] ...
121 >     [[terminal6]] ...
129 >     [[terminal7]] ...
137 [plugins]
138 [profiles]
139     [[default]]
140         cursor_color = "#aaaaaa"
141     [[collabot_set]]
142         cursor_color = "#aaaaaa"
143         exit_action = hold
144         foreground_color = "#ffffff"
145
```

layout 탭에는 **collabot** 과 **collabot_opt** 레이아웃이 저장되어있음.

collabot 레이아웃 : 구버전 책장 인식 모듈 - SSIM

collbot_opt 레이아웃 : 현재 버전 책장 인식 모듈 - yolov5 and optical flow

```
[[collabot_opt]]
[[child0]]~
[[child1]]~
[[child2]]~
[[child5]]~
[[terminal3]]
  command = source /home/kist/collabot_t.bash; roslaunch collabot_do run_optical_flow.launch;bash
  order = 0
  parent = child2
  profile = collabot_set
  title = Main
  type = Terminal
  uuid = 8f77accl-55c8-41b2-98b4-e467fcbc5d88
[[terminal4]]
  command = source /home/kist/collabot_t.bash; sleep 1s; rosrund serial_python serial_node.py _port:=/dev/ttyACM0;bash
  order = 1
  parent = child2
  profile = collabot_set
  title = OpenCR
  type = Terminal
  uuid = 5862c8a7-63d4-4cc9-915e-12196c1b9cca
[[terminal6]]
  command = source /home/kist/collabot_t.bash; sleep 1s; roslaunch az_body_tracker run.launch;bash
  order = 0
  parent = child5
  profile = collabot_set
  title = Body_tracker
  type = Terminal
  uuid = ca47c9a4-3e5b-46bc-b54f-5f399e0e3b19
[[terminal7]]
  command = source /home/kist/collabot_t.bash;conda activate collabot;cd ~/yolov5;python detect_book_state.py;bash
  order = 1
  parent = child5
  profile = collabot_set
  title = Drawer_detector
  type = Terminal
  uuid = 6ddf5566-cced-42af-b475-3558c81ab173
```

레이아웃마다 4개의 하위 터미널 구성이 있으며, 해당 터미널 하위에 실행 command를 지정 가능.

tconfig에서 수정사항이 생겼을 경우, 저장 후 모든 터미네이터 종료 후 실행 해야 함.

분할 터미널 소스파일

runopt 실행 시 생성되는 터미널은 ~/.bashrc를 소스하지 않고, ~/collabot_t.bash를 소스함.

/collabot_t.bash에는 아래와 같이 ROS 소스, ROS 마스터 , 터틀봇, conda 에 관한 내용으로 구성.

~/bashrc에서 ROS 마스터 , 터틀봇, conda 관련 수정사항은 collabot_t 배쉬에도 반영해야 함.


```
collabot_t.bash
~/

Open ▾

### source ROS ###
source /opt/ros/melodic/setup.bash
source ~/catkin_ws1/devel/setup.bash

### turtlebot ###
export TURTLEBOT3_MODEL=waffle_pi
export OPENCV_MODEL=burger
export OPENCV_PORT=/dev/ttyACM0

### ROS MASTER SET ###
export ROS_PACKAGE_PATH=~/catkin_ws1/src:/opt/ros/melodic/share

#export ROS_MASTER_URI=http://192.168.0.100:11311
#export ROS_HOSTNAME=192.168.0.100

export ROS_MASTER_URI=http://172.16.0.75:11311
export ROS_HOSTNAME=172.16.0.75

### conda setup ###

# >>> conda initialize >>>
# !! Contents within this block are managed by 'conda init' !!
__conda_setup="$(('/home/kist/anaconda3/bin/conda' 'shell.bash' 'hook' 2> /dev/null)"
if [ $? -eq 0 ]; then
    eval "$__conda_setup"
else
    if [ -f "/home/kist/anaconda3/etc/profile.d/conda.sh" ]; then
        . "/home/kist/anaconda3/etc/profile.d/conda.sh"
    else
        export PATH="/home/kist/anaconda3/bin:$PATH"
    fi
fi
unset __conda_setup
# <<< conda initialize <<<
```

책장 인식 모듈

책장 인식에 yolov5, 책 이탈 여부 판정에 optical flow 적용됨.

모듈 경로 : ~/yolov5

메인 스크립트 : ~/yolov5/detect_book_state.py

>> 해당 스크립트에서 책장 인식시간, 책 이탈 판정 시간 수정 가능.