# Instructions for jacket removal

## Launch terminals:

in all terminals

```
cd Desktop/PierreROS
./baxter.sh
```

terminal 1

```
roslaunch irex init.launch
```

terminal 2

```
roslaunch irex init_project.launch
```

terminal 3

```
roslaunch jacket_remover main.launch
```

## To launch project:

for phase 1 and 2:

we need a laptop that is connected to the realsense camera and connected to the wifi with ethernet cable

run final_clust.py on laptop
run grip.py on baxter computer

for phase 3:

run 3d_traj.py on laptop
run final_phase_bax.py on baxter computer

## Explanations:

I used two computers for the project due to the fact that the Baxter computer runs on ubuntu 14 and therefore i wasn't able to use the newest libraries and programs for the vision scripts

final_clust takes in the information from the realsense camera and sends to the baxter computer the midpoints of the center of the two pieces of tape on the jacket

grip.py reads these midpoints and tries to grab them, once it has grabbed the points it moves behind the shoulders of the person. It uses socket to read the midpoints, however this method is prone to errors since the messages sent by socket are often incomplete.

3d_traj.py reads the information from the camera and uses mediapipe to skeletonize the person in front of baxter. With the skeleton information it creates a 3d line along the arms of the person. it saves the trajectories in a list

final_phase_bax.py takes in 4 vectors that describe the direction of the trajectory along the arms. We then use these vectors to move the arms of Baxter along the trajectory. The movement of the arms is done in a loop in order to slow down the speed of the movement in case there is a collision with the subject. I did it this way since I wasn't able to properly calibrate the system. Ideally we would directly use coordinates from the skeleton and order Baxter to move to these different coordinates.

**Problems:**

At the moment the program is not functional due to the calibration problem. Indeed, we are not able to grab the jacket properly because the camera and Baxter are not coordinated well.

Communication between the two computers doesn't work very well and sometimes the messages sent are not complete which causes errors in the programs

**Improvements to make:**

Fix calibration using the method described in Ravis thesis?

Find way to control speed of Baxter in order to make it safe in case of collisions

If we want to control each arm independently we need to make one script for each arm and run them at the same time instead of controlling both arms in the same script

A new way to communicate between the two computers needs to be used because sending information using socket is not reliable. Using ROS publishers and subscribers to send the information would probably be a better choice. Ideally we would only want to use one computer for everything.

in grip.py:
phase 2 needs to be corrected since it doesn't take into account information of the subject. We need to use the skeleton information to be aware of the localization of the subjects shoulders and use this information to avoid coming into contact with the subject when moving the arms

in final_phase_bax.py:
At the moment, I am using vectors to generate the coordinates in Baxters reference frame to move the Baxter arms. This is not ideal because it is not adaptive. Once the calibration problem is fixed, we can use the skeleton information and the trajectory generated in the realsense camera reference frame and transfer it to the Baxter reference frame. Baxter then just has to move along this trajectory to remove the jacket.