17/10/21

Interfacing

Things done:

- Objects/Human detection and classification on the L515 using YOLOv5 and deepsort()
- Able to get the number of detected objects in the frame (L515)
- Able to get the range of the detected objects using L515 through codes

- Able to pull point cloud data (azimuth, elevation, range, x, y, z, etc) and store them as objects for the IWR6843

- Servo motor is working

- Characterization of the sensors

Current state of project:

- Currently every device is able to perform independently

- Due to our oversight of the specs requirement of the mmwave SDK, a laptop is required to be integrated into our system as an additional device because the jetson nano could not support the mmwave SDK. However, should this project be replicated, the laptop can be removed by finding a microcontroller that supports 32-bit software.

- Due to the above point, current algorithm is as follows:
  1. The mmWave sends the detected point clouds/clusters to the laptop
  2. The location of the point clouds is parsed on the laptop and the signal to activate the servo motor is sent from the laptop to the Jetson Nano through putty
  3. The jetson Nano activates the servomotor which pans and tilts the L515 towards the cluster.
  4. The human detection, ranges and other information are performed on the Jetson Nano and sent back to the laptop to display whether any rules are violated.

  *should this project be replicated, the laptop is redundant as long as the microcontroller can install 32-bit software

- To maximise the performance of the IWR6843, raw data form the sensor is required and hence, the $500 DCA1000 board is ordered

- Algorithm to obtain the distance between two targets from the L515 has been designed which requires the calibration of the 3d camera.

Problems faced:

- As mentioned, the 64-bit jetson nano does not allow for 32-bit software, hence an additional dimension (laptop) of interfacing is introduced to our system.

- This prevents the set up to be not as portable as our previously intended set up
- However, this allows for a more accurate and better performing AI algorithm which is the YoloV5 and deepsort model. If this project is to be replicated without the laptop, and the microcontroller is unable to support the 2 neural networks, mobilenet ssd should be implemented instead which requires much lesser memory and computational power but it is also slower and less accurate than YOLO

- Point cloud data from the IWR6843 is insufficient, we have to dwell into the source code of the mmwave chip itself to decrease the threshold of the velocity for the detected points.

- We need the raw data of the IWR6843 in order to maximise the performance of the mmwave sensor, hence, we require another $500 board (DCA1000)

Work to be done:

- Writing a script on the laptop to run putty remotely

- Configure the detection settings of the IWR6843 and lower the velocity threshold in order to obtain more detections

- Study the data obtained from the tests we conducted in the lab in order to determine the performance of the sensors and the setup