Novel Object Detection, Classification and Localization (NODCAL)

Austin Shin, Raphael Norman-Tenazas, Thomas Keady
3/27/19

Description:
Given a map of the environment, a robot will determine novel objects in its environment using a LIDAR, estimate their position and velocity with error, classify them using a camera, and determine its actions based on its classification. This will be a generalizable node that will output the positions, velocities, errors and classifications of the objects. As an example, we will demonstrate using another robot colored with different markings. Our robot will use the classification to run away from a robot colored with orange and towards the other robot colored with blue. The colored robot will be remote-controlled by a user while the other is autonomous. And if time permits, we will look towards incorporating multiple other objects as opposed to just one other.

Software:
New packages:
- nodcal
    - Described above
    - Subscribes
        - scan
        - depth/image_raw
        - depth/points
    - Publishes
        - detected_objects (custom msg)
- Nodcal_example_actions
    - Subscribes
        - detected_objects
    - Publishes
        - cmd_vel

Existing packages:
- realsense_camera
    - (RNT) Used this package before and was able to display image and color depth points in RVIZ
- amcl

- - For building the map and navigating within it. Will use point cloud output to detect novel objects in separate node

Hardware and Infrastructure:
- New Hardware
  - 1 x Slamtec A2M8 RPLIDAR
- Existing Hardware
  - 2 x Turtlebot 3
  - 1 x Realsense camera
  - 1 x ODROID XU4
- Previous Testing
  - (RNT) Heavy testing with all products. Build TB3 from scratch with up board and integrated realsense camera
  - (AJS) Worked with RGB images, depth map, and point cloud from Realsense D435, which is very similar to R200 included in RDK
  - (TMK) Used A2M8 LIDAR for 2D localization in known map, both simulated and deployed
- Purchasing Details
  - Slamtec A2M8 RPLIDAR
    - Already on-hand

Safety Plan
- Risks
  - None (bruised toes from Turtlebot)
- Plan for Management and Mitigation
  - None (wear shoes)

Timeline
- Week 1
  - Software
    - Install ROS, dependant packages on UP board
    - Camera calibration
  - Hardware
    - Build TB3, attach sensors
  - Testing
    - Visualize sensor data in RVIZ
- Week 2
  - Software
    - Write demo node

- ■ Write custom message file
  - ○ Hardware
    - ■ Design / 3D print holder for camera to attach to TB chassis
  - ○ Testing
    - ■ Given custom messages, test actions from demo node
- ● Week 3
  - ○ Software
    - ■ Write algorithm for novel object detection and classification
  - ○ Hardware
    - ■ N/A
  - ○ Testing
    - ■ Simulation
- ● Week 4
  - ○ Software
    - ■ Write algorithm for position and velocity estimation
  - ○ Hardware
    - ■ N/A
  - ○ Testing
    - ■ Simulation
  - ○ Other
    - ■ Start writing project report and making poster
- ● Week 5
  - ○ Software
    - ■ Incorporate previous classification and estimation in making decision about action
  - ○ Hardware
    - ■ N/A
  - ○ Testing
    - ■ Simulation
  - ○ Other
    - ■ Final edits of project report and poster