NOVEL Object Visualization, Evaluation and Localization (NOVEL)

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

Project Advisor: Gabriel Baraban

Description:

Given a map of the environment (which will be a plywood imitation of the DJI Robomaster field placed in Malone G35), a robot will determine novel objects in its environment using a LIDAR, estimate their position (and velocity if time permits) with error, classify them using a camera, and determine its actions based on its classification. We define a novel object as an object that was not previously in the known map that the robot has of the environment.

With a known map of the environment, approximate LIDAR measurements from the robot can be calculated given its position. As the robot moves autonomously, it will compare its observed LIDAR measurements with the its calculated LIDAR measurements using cross-correlation. If there is a significant error, the robot will further process the data by determining which measurements cause the error using one-dimensional filtering techniques. Using this information, the number of objects and an approximate angle and distance can be calculated. We will also compute an uncertainty for these values.

After an object is detected using the LIDAR, the camera will then be oriented such that the novel object detected is within its field of view. The robot will then perform classification using convolutional neural networks, such as YOLO. It will also unify depth information from the camera to reduce the uncertainty on the position using a Kalman filter. If time permits, we will add additional functionality to determine the velocity of the object using object tracking.

Depending on what the classified object is and how it is moving, the robot will then perform a certain action. The following scenario will explain more about these steps.

As an example, we will have a demo using one robot and other commonplace objects. The robot will be autonomous and controlled by our software. There will be a LIDAR and an Intel Realsense R200 camera as the sensors. The robot will know the map of the environment it is in, and the user will designate a target location the autonomous robot should move to. As it moves to this location, using the LIDAR, it will begin to detect whether there are unknown objects creating large discrepancies in its LIDAR measurements. After detecting an object, the autonomous robot will reorient its camera to put the object in the camera's field of view, and using the image from the camera, it will classify the object. The autonomous robot can also approximately determine the location and movement of the novel object, and it will changes its trajectory appropriately. If time permits, we will look towards incorporating multiple other moving objects as opposed to just one other.

Here are the goals we want to achieve along with some evaluation metrics during the following weeks in order (though some can be worked on in parallel):

- Detecting up to 3 stationary novel objects using LIDAR and rotate camera appropriately
    - Metric: time it takes to classify and localize robot once it is in view/line of sight (starting baseline: 2 sec to identify using LIDAR, 3 sec to orient camera and perform classification)
- Classify and localize stationary novel objects in camera image
- Plan path from any arbitrary starting point to arbitrary end point in known environment, and move along it without any novel objects
- Adjust path based on stationary novel object location in map such that the robot can most accurately localize the detected novel objects
- Estimate velocity of moving novel object detected
- Adjust path to most accurately localize moving novel objects

Software:
New packages:

- nodcal
    - Described above
    - Subscribes
        - scan
        - depth/image_raw
        - depth/points
    - Publishes
        - detected_object (custom msg)
- camera_control
    - Controls camera to put detected object in field of view
    - Subscribes
        - detected_object
- Nodcal_example_actions
    - Subscribes

        - detected_objects
        - object_vels
    - 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
    - 1 x Turtlebot 2 with Thinkpad Yoga 260
    - 1 x Realsense R200 camera
- 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
- Infrastructure
    - Mock-up of DJI RoboMaster competition field (5m x 8m) made of plywood
        - Can be set up in Malone G35, room reserved on a weekly basis

Safety Plan

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

Timeline

- Week 1
  - Software
    - Ensure camera and LIDAR are working and can obtain necessary data from them
    - Create environment in RVIZ and Gazebo
  - Hardware
    - N/A
  - Testing
    - Visualize camera + LIDAR data in RVIZ
  - Other
    - Finish editing proposal and resubmit
- Week 2
  - Software
    - Write node for object detection based on LIDAR
    - Write node object classification/localization based on Realsense images
    - Create xacro model of robot
  - Hardware
    - Design / 3D print holder for camera to attach to robot chassis
  - Testing
    - Test LIDAR object detection in simulation
      - Compare single object detection performance versus multi-object detection (up to 3 novel objects)
    - Test object classification/localization using actual images from Realsense camera
- Week 3
  - Software
    - Write node for controlling camera orientation based on LIDAR and integrate LIDAR detection with camera classification/localization
    - Write action node for deciding what action to take based on detected object and its location
  - Hardware
    - N/A
  - Testing
    - Evaluate how long it takes to classify and localize novel object
- Week 4
  - Software

        - Incorporate path planning capability so that robot can navigate from any arbitrary starting point to arbitrary end point
        - Camera calibration to estimate object position relative to camera coordinate frame
    - Hardware
        - N/A
    - Testing
        - Test path planning capability in RViz (similar to assignment 7) and using Turtlebot
    - Other
        - Start writing project report and making poster
- Week 5
    - Software
        - Enhance path planning package by incorporating detected object and its velocity
        - Add object velocity estimation to node that performs classification/localization
    - Hardware
        - N/A
    - Testing
        - Integrate all parts of the project together and ensure the whole pipeline works together
    - Other
        - Final edits of project report and poster