Advanced Control Laboratory (085705)
ROS #3: Projective Geometry Using Real-Data

This lab focuses on aspects of projective geometry using real-data, recorded from the Pioneer mission at ANPL.

Everything should be submitted in a single PDF file to the Moodle course cite.

1. (40%) Run the Pioneer mission and record the data.
   (a) (5%) Calculate and present the waypoints (Start,WP_1,WP_2,WP_3, Target) for the robot using the supplied scenario "ACL exp layout.pdf" given in Moodle. Each waypoint should be presented in the form of a 3D point, the heading angle and the equivalent rotation quaternion.
   (b) (35%) Update your main function (see lab ROS#2, Q.2) to do the following:
      i. Contain the aforementioned waypoints.
      ii. Command the robot to visit each of the waypoints (by order).
      iii. Store the path traveled by the robot using concatenation of the information from the parameter "robot_path", be sure to keep it with time indices. Remember that "robot_path" information originates from odometry readings.
      iv. Save the path traveled by the robot, again with time indices.
      v. At the end of the run, plot the trajectory traveled by the robot (using just odometry readings).
      vi. Change the velocity and pose topics to '/RosAria/cmd_vel' and '/RosAria/pose' accordingly, so you'll be able to communicate with the actual pioneer robot rather than the Gazebo simulation.
   (c) Run the Pioneer mission in ANPL and record the data.
2. (10%) Calibrate the Pioneer's Camera.
   (a) Download the camera calibration toolbox for Matlab[1] .
   (b) Use the captured calibration pattern images (supplied in Moodle) and follow the described steps[2] to calibrate your own camera.
   (c) Submit snapshots of the process.
   (d) Write an expression for the estimated camera calibration matrix K. Indicate the principal point, and focal length in each axis.
3. (50%) Projection of 3D points
   (a) Pick an image from the recorded data, that shows both landmarks, and display the image in Matlab (using the command `image`).
   (b) (15%) Calculate the location of each Landmark with respect to the camera using ground truth.
   (c) (5%) Extract manually the image coordinates that correspond to each Landmark (you can use the Matlab command `ginput` for that). Display on the image these image coordinates.
   (d) (20%) Project the calculated Landmark 3D locations onto the camera frame using the measured locations (from section 3.b) and the camera calibration matrix (from section 2.d). Indicate the projected image coordinates and display them on the image.

[1] http://www.vision.caltech.edu/bouguetj/calib_doc/.
[2] http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/example.html.

(e) (10%) Calculate the re-projection error for each Landmark by comparing between the extracted image coordinate and the projection of the corresponding Landmark's 3D point.