Project 2: Particle Filter SLAM

*Collaboration in the sense of discussion is allowed, however, the assignment is **individual** and the work you turn in should be entirely your own. See the collaboration and academic integrity statement here:* https://natanaso.github.io/ece276a. *Books, websites, and papers may be consulted but not copied from. It is absolutely forbidden to copy or even look at anyone else's code. Please acknowledge **in writing** people you discuss the project with and provide references for any books or papers you use.*

Submission

Please submit the following files on **Gradescope** by the deadline shown at the top right corner.

1. **Programming assignment**: upload all code you have written for the project (do not include the provided datasets) and a README file with a clear, concise description of the main file and how to run your code.

2. **Report**: upload your report in pdf format. You are encouraged but not required to use an IEEE conference template[1] for your report.

Problems

In square brackets are the points assigned to each part.

1. [40 pts] Implement simultaneous localization and mapping (SLAM) using odometry, 2-D LiDAR scans, and stereo camera measurements from an autonomous car. Use the odometry and LiDAR measurements to localize the robot and build a 2-D occupancy grid map of the environment. Use the stereo camera information to add RGB texture your 2-D map.

   - **Data**: available here:
     - param.zip: https://drive.google.com/file/d/1alGT4ZJcNKCLLLW-DIbt50p8ZBFrUyFZ/view?usp=sharing
     - sensor_data.zip: https://drive.google.com/file/d/1s82AV_TACqqgCJaE6B561LL1rgWmewey/view?usp=sharing
     - stereo_images.zip: https://drive.google.com/file/d/1kJHOm9-1Zz13rUBg46D3RS1TOeXZEhQS/view?usp=sharing
   - The vehicle is equipped with a variety of sensors. In this project, we will only use data from the front 2-D LiDAR scanner, fiber optic gyro (FOG), and encoders for localization and mapping as well as the stereo cameras for texture mapping. See Fig. 1 for an illustration.
     - All parameters and static transformations can be found in param.zip. All transformations are provided from sensor to the body frame, i.e., $_BT_S \in SE(3)$.
     - The FOG provides relative rotational motion between two consecutive time stamps. The data can be used as $\Delta\theta = \tau\omega$, where $\Delta\theta$, ω, and τ are the yaw-angle change, angular velocity, and time discretization, respectively. This article provides information about how a FOG works.
     - The sensor data from the 2D LiDAR, encoders, and FOG are provided in .csv format. The first column in every file represents the timestamp of the observation. For the stereo camera images, each file is named based on the timestamp of the picture.
   - The goal of the project is to use a particle filter with a differential-drive motion model and scan-grid correlation observation model for simultaneous localization and occupancy-grid mapping. Here is an outline of the necessary operations:
     - **Mapping**: Try mapping using the first LiDAR scan and display the map to make sure your transforms are correct before you start estimating the robot pose. Remove scan points that are too close or too far. Transform the LiDAR points from the LiDAR frame to the world frame. Use bresenham2D or cv2.drawContours to obtain the occupied cells and free cells that correspond to the LiDAR scan. Update the map log-odds according to these observations.

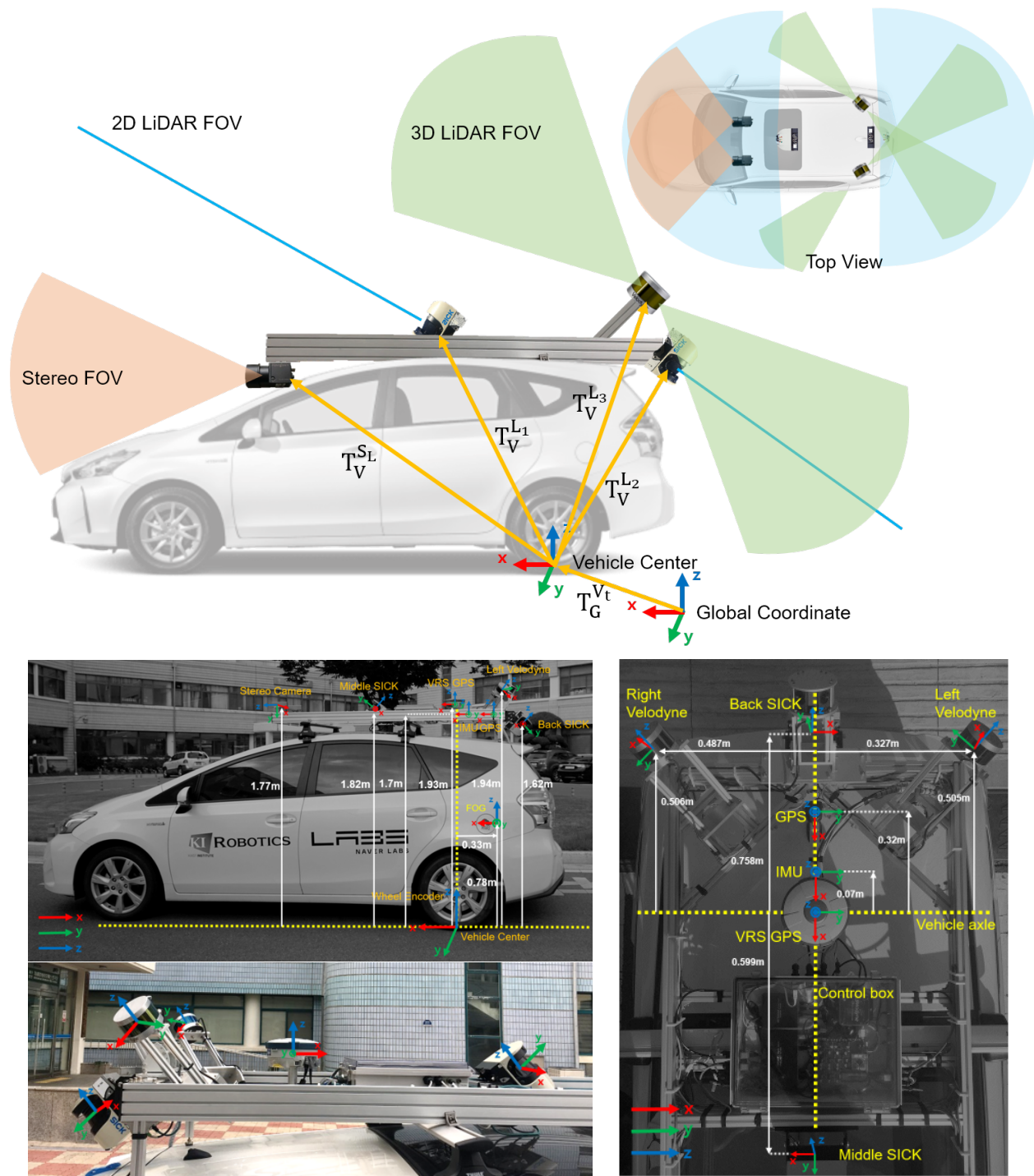[1]https://www.ieee.org/conferences_events/conferences/publishing/templates.html

Figure 1: Sensor layout on the autonomous vehicle. We will only use data from the wheel encoders, fiber optic gyro (FOG), front 2D LiDAR (Middle SICK), and the stereo cameras. See param.zip for the parameters and static transformations for each sensor.

– **Prediction**: Implement a prediction-only particle filter at first. In other words, use the encoders and the FOG data to compute the instantaneous linear and angular velocities v_t and ω_t and estimate the robot trajectory via the differential drive motion model. Based on this estimate, build a 2-D map before correcting it with the LiDAR readings. In order to see if your prediction step makes sense, try dead-reckoning (prediction with no noise and only a single particle) and plot the robot trajectory.

– **Update**: Once the prediction-only filter works, include an update step that uses scan-grid correlation to correct the robot pose. Remove scan points that are too close or too far. Try the update step with only $3 - 4$ particles at first to see if the weight updates make sense. Transform the LiDAR scan to the world frame using each particle's pose hypothesis. Compute the correlation between the world-frame scan and the occupancy map using mapCorrelation. Call mapCorrelation with a grid of values (e.g., 9×9) around the current particle position to get a good correlation (see p2_utils.py). You should consider adding variation in the yaw of each particle to get good results.

– **Texture map**: Compute a disparity image from stereo image pairs using the provided script in p2_utils.py and estimate depth for each pixel via the stereo camera model. Project colored points from the left camera onto your occupancy grid in order to color it. Determine the depth of each RGB pixel from the disparity map and transform the RGB values to the world frame. Find the plane that corresponds to the occupancy grid in the transformed data via thresholding on the height. Color the cells in the occupancy grid with RGB values according to the projected points that belong to the its plane.

2. Write a project report describing your approach to the SLAM and texture mapping problems. Your report should include the following sections:

- [5 pts] **Introduction**: discuss why the problem is important and present a brief overview of your approach

- [10 pts] **Problem Formulation**: state the problem you are trying to solve in mathematical terms. This section should be short and clear and should define the quantities you are interested in precisely.

- [35 pts] **Technical Approach**: describe your technical approach to SLAM and texture mapping.

- [10 pts] **Results**: present your results, and discuss them – what worked, what did not, and why. Make sure your results include (a) images of the trajectory and occupancy grid map over time and (b) textured maps over time. If you have videos, include them in the zip file and refer to them in your report.