16-662: Robot Autonomy
Spring, 2015

Project 2B Description
Due: March 16

Overview and Objectives

Project 2B considers stereo visual odometry and Kalman filtering. The project seeks to:

- implement a stereo visual odometry algorithm
- implement an Unscented Kalman Filter

Project Report Format and Grading Criteria

Each group is expected to submit a two page (or more) project report that will be graded out of a total of 10 points based on content correctness (4), quality (2), completeness (2), and clarity of presentation (2). Consider the following representative questions when preparing the report.

- Correctness: Is there a notable or unexplained error or phenomena presented in the results not addressed by the discussion? Do the results correspond to the project topic and adequately support the discussion claims? Is there sufficient statistical evidence to show that the results are repeatable and valid?
- Quality: Do the results present an acceptable level of performance (e.g., error) and is there a satisfactory explanation as to why the performance is acceptable given the system characteristics and methodology?
- Completeness: Do the results and discussion adequately address the full problem with consideration of the multiple problem facets? Are there any notable omissions in the approach, discussion, or results presentation?
- Clarity of presentation: Is the discussion cohesive, coherent, devoid of grammatical errors, and self-contained? Do the figures and tables adequately present the data? Are the figures labeled and easily read when printed with appropriate color (or grayscale) schemes, line thickness, and structural formatting?

Useful Information

The images in part 1 of this project have already been rectified. The stereo baseline is 0.1621 m. The intrinsic parameters are

K11: 164.255034407511
K12: 0.0
K13: 214.523999214172
K21: 0.0
K22: 164.255034407511
K23: 119.433252334595
K31: 0.0
K32: 0.0
K33: 1.0

Project Report Content

1. Stereo Visual Odometry:
    a. Take the first pair of left-right images and extract features (e.g. BRISK, FAST, Harris, MSER, SURF) in them. You may use feature detection functions found in the MATLAB Computer Vision Toolbox.
    b. Match features between the left and right image to obtain a set of stereo correspondences. You may use MATLAB's matchFeatures function.
    c. Using the stereo correspondences, triangulate the real world feature locations in a reference frame centered at the left camera's origin (assuming a pinhole model). Include a 3D plot depicting the feature locations and the labelled camera axes.
    d. Find feature correspondences between the left image and the left image at the next timestep. Compute the incremental rotation and translation between the successive left camera reference frames by minimizing reprojection error on the second left image. These steps should be incorporated into a RANSAC routine to reject outliers. You may use MATLAB's generalized RANSAC routine, or you may write your own. Visualize the corresponding points (before and after outlier rejection) with MATLAB's showMatchedFeatures function.
    e. Repeat steps a-d for all images in the dataset to find incremental translations/rotations between all successive timesteps. Assume that the camera starts at the origin of the world reference frame with an identity orientation. Compute the full trajectory (in the world reference frame) by composing the sequence of estimated incremental pose transformations. Include a plot of the position and orientation (Euler Angles) vs. time, as well as a 3D plot of the position.
    f. Repeat step e while varying RANSAC parameters and describe the effect on overall algorithm performance.
    g. From the image frames, it is apparent that the camera returns to the same location that it started at. Does the trajectory computed by the implemented stereo visual odometry algorithm return to its starting location? Explain the presence/absence of drift in the estimated trajectory. Bonus: Implement a strategy to reduce drift that does not relate to RANSAC (e.g. discarding every other pair of images) and explain how it reduces drift.
2. Unscented Kalman Filter
    a. Fill in the provided MATLAB function imuProcessDynamics to implement the state propagation model. This function takes in the current vehicle state, noise vectors, and IMU measurements (angular velocity and linear acceleration) and outputs the vehicle state at the next time step.
    b. Fill in the provided MATLAB function getSigmaPoints to compute the sigma points for a given mean vector and covariance matrix.
    c. Fill in the provided MATLAB function processUpdate to propagate sigma points through the process dynamics and compute the prior state mean and prior covariance.
    d. Fill in the provided MATLAB function absoluteCorrection to propagate sigma points through the observation function and compute the posterior state mean and covariance.
    e. Tune the noise parameters. How does changing the initial state covariance values (i.e. the init_X_sigma parameters) impact the state estimate? What is the impact of changing the relative magnitudes of the IMU model uncertainty

parameters (sigma_{w,bw,a,ba}) and the exteroceptive sensor model uncertainty parameters (sigma_{xy,z,yaw})? Provide a few comparison plots to illustrate the effects of changing the noise parameters. Describe the strategies/heuristics used to obtain the noise parameter values and write down the set of noise parameters. Use this set of parameters for parts f and g.

f. The ground truth position, velocity, and orientation (in ZYX Euler angles) have been provided in hand_carry.mat. Compute the error of the UKF estimated position with respect to the ground truth. You will need to interpolate from the ground truth timestamps to the UKF timestamps before subtracting. Note that some parts of the ground truth data are highly oscillatory due to occlusions leading to loss of tracking in the vicon system.

g. Select a time interval over which the ground truth position is qualitatively smooth and plot the position error vs. time. Next, make a histogram plot of the position errors over the selected time interval. Find the empirical standard deviation of the position error over the selected time interval and compare it to the estimated UKF 1-sigma values (roughly approximated by the square roots of the first 3 diagonal elements in the state covariance matrix). Explain why the empirical standard deviation is larger/smaller/equal to the UKF estimated uncertainties.