Device Movement Detection

STUDENTS: Aaron Lin, Jerry Chen, Pei-Hsuan Lin, Ryan Ching, Xinqi Chen, Yuanhao Luo, Zhengyang Li

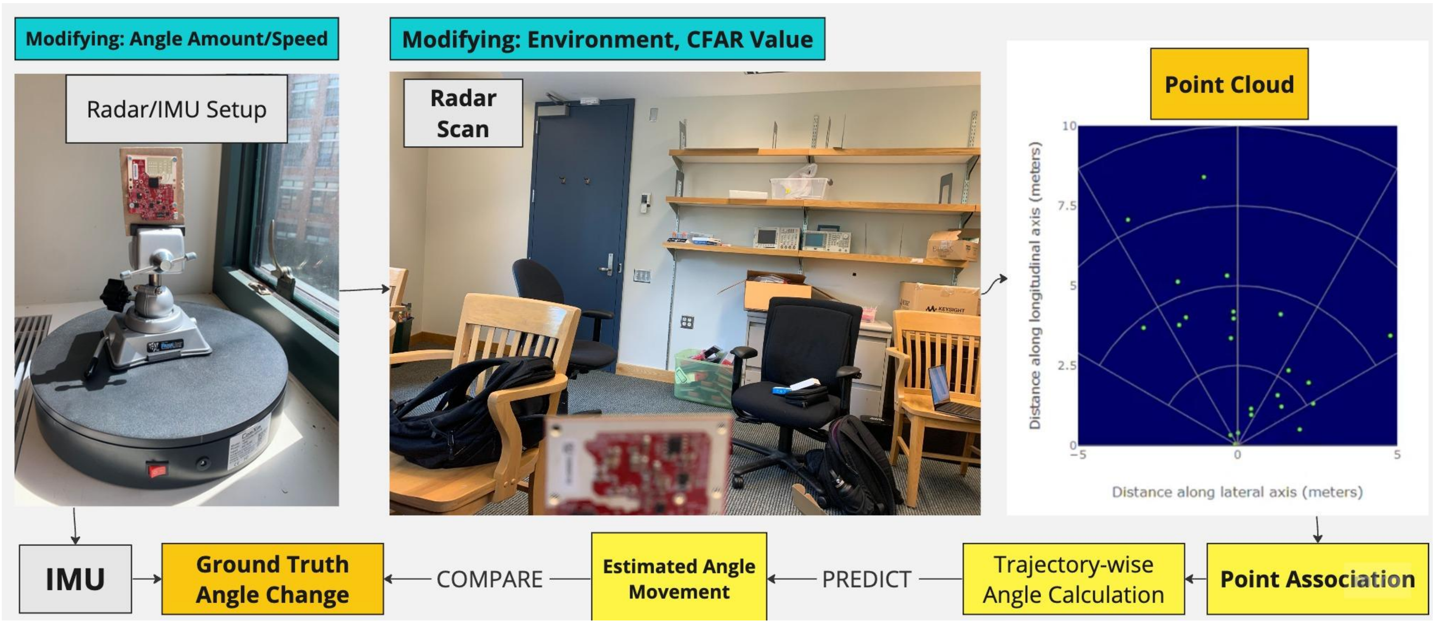
amazon Lab126

26

Introduction

- The device movement detection project aims to develop an algorithm that can detect and track the movement of the desired device (eg: Amazon Halo Rise) using MM-wave radar.
- The project involves preprocessing the point cloud from radar data at each frame to remove noise and then applying a motion detection algorithm to identify the existence of the movement as well as the exact rotating angle and translational distance. The algorithm then tracks the feature points across frames, using techniques such as nearest neighbor association and KD-Tree to improve tracking accuracy. An IMU is used as ground truth for result verification.

Environment Setting, Data Processing, Design Overview



Environment Setting
Data is collected by rotating our radar around the platform shown in Radar/IMU Setup. We set up our environment and measurement using the following criteria:
- CFAR threshold: Affects number of points collected by radar.
- Obstacles: Tested environments with high/small number of objects
- Speed: Speed of rotation device when measuring points
- Angle: Device rotation amount

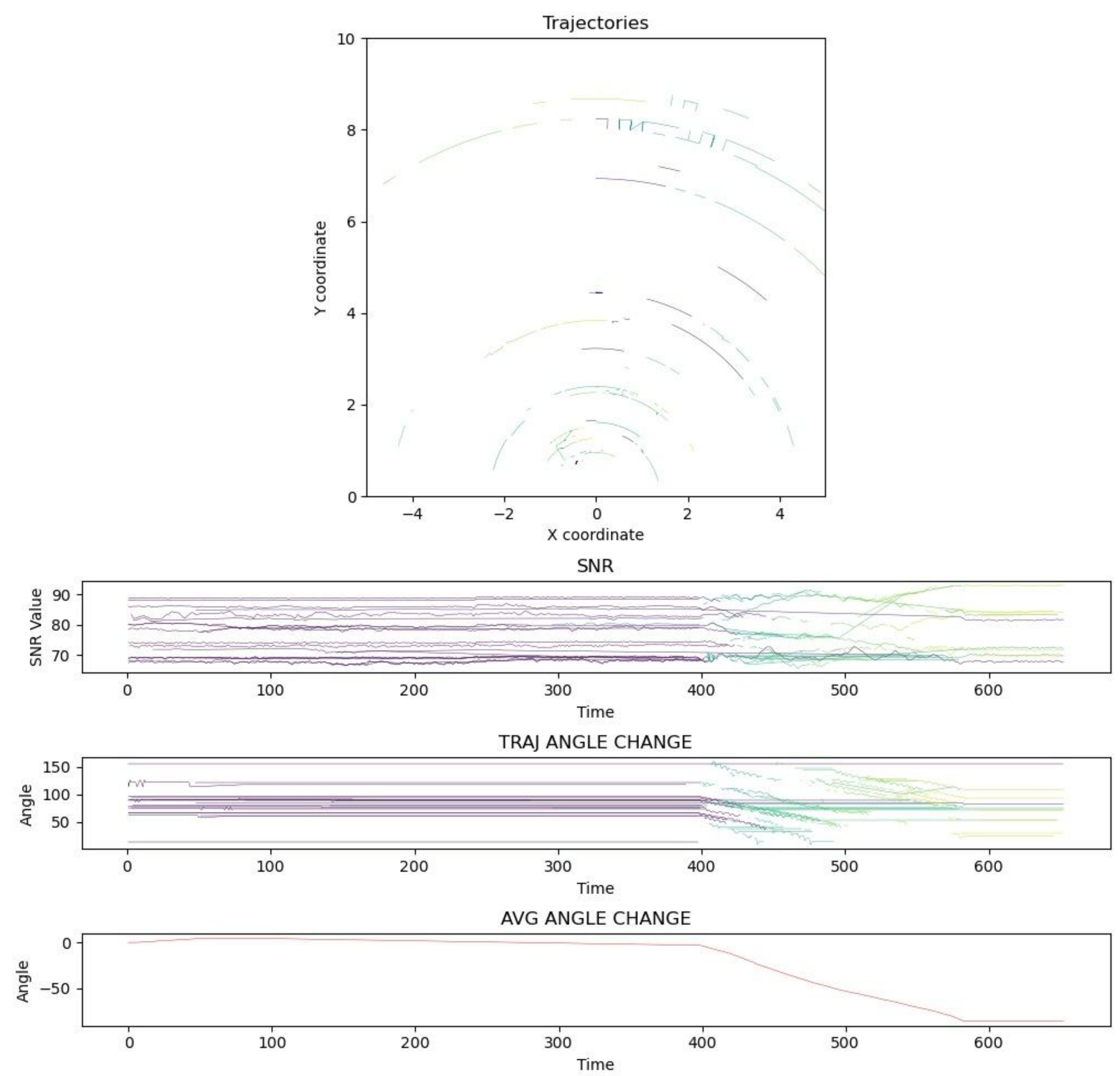
IMU
- An Inertial Measurement Unit is used to measure the ground truth of rotational movement, used to compare our measured angle to calculate error.

Design Overview
- Radar Scanning: An MM-wave radar scans environment marking objects in a 120° field of view as a point, recording the X-Y Position relative to radar, and Signal-to-Noise Ratio. The output is a CSV file with this information.
- Point Association: Associate points from consecutive radar scans. Finds associations between points in subsequent frames to track objects over time (See Point Association Section)
- Estimated Angle Movement: Given the angle change of all objects, use averaging to find the estimated angle movement (see Movement Calculation)

Point Association

Data Processing:
- An Inertial Measurement Unit is used to measure the ground truth of rotational movement.
- The radar scan information is outputted to a .csv file.

Association:
- After receiving a point cloud (described in our data processing step), we aim to use the movement of the points across frames to determine rotation amount.
- Begin by using all points in the first time step as the beginning of a trajectory
- Associate points across time using the following thresholds:
  - **Distance:** Radar operating at 30 FPS, subsequent points will be spatially close.
  - **SNR:** Points belonging to the same object read consistent SNR values
  - **Time:** Due to noise, some points may pop in and out of frames, this threshold handles reappearing points
- For each point in our trajectory list, filter all points in the next frame by the three thresholds as described above, appending the closest point for each trajectory.
- Create new trajectories for all points that are not mapped to existing trajectories.
- Discard short trajectories (under 5 points), likely due to noise as points corresponding to objects are expected to have many points.
- The figure below illustrates the estimation process of a 90 degree rotation.

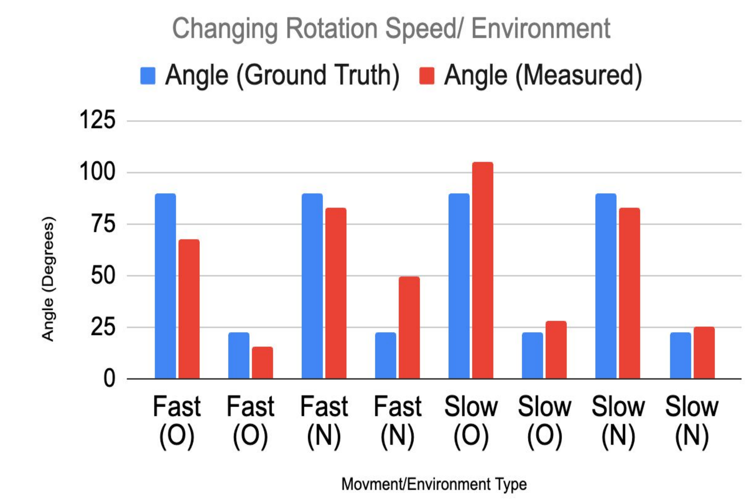


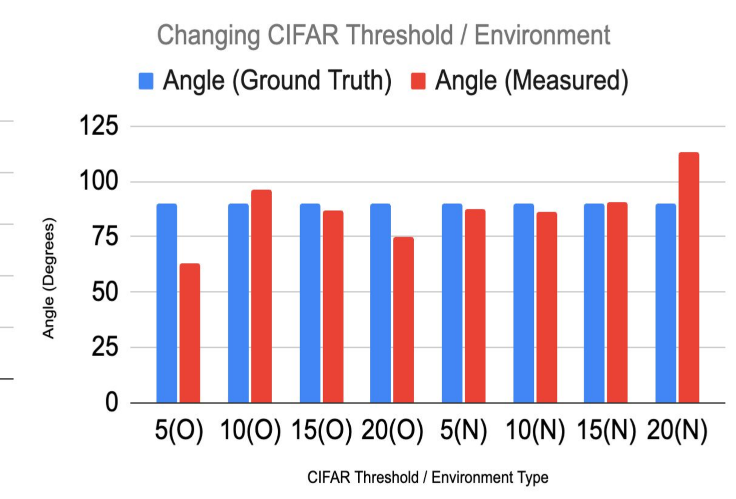
Movement Calculation

- Once we have collected all trajectories, we will measure the angle change throughout all time steps, shown in TRAJ ANGLE CHANGE plot.
- Next, we apply smoothing to each path, taking the average angle change at the start and end time step of when each trajectory angle begins to change. Due to noise, there may be drastic angle changes or no changes as our radar is not perfectly accurate, which would affect the result in our final step. For each timestep, we store the average angle change at that point in time for all trajectories.
- Repeating this process for all frames, we then take the average angle change for each time step, which leads to our result shown in the AVG ANGLE CHANGE plot.
- The angle at the final timestep of average angle change is our estimated rotation.

Results Analysis

- Generally, higher CFAR Thresholds tended to work better on high obstacle environments and a lower CFAR Threshold worked better on no obstacle environments.
- The figures below illustrates how our algorithm performs varying the CIFAR threshold and rotation speed in high/low obstacle environments. O/N represents an environment having obstacles/no obstacles, respectively.





Future Work, References, and Acknowledgments

- Dynamic CIFAR Thresholding to determine optimal value based on number of points received and speed of rotation.
- Machine Learning to ignore noisy or flickering points.
- Combination of radars to increase field of view for large rotations.

Faculty/Mentors: Mingfei Chen, Prasad Shamain, Mahmood Hameed.
Graduate Students: Pei-Hsuan Lin, Yuanhao Luo, Aaron Lin, Ryan Ching
Undergraduate Students: Jerry Chen, Xinqi Chen, Zhenyang Li

S. H. Cen and P. Newman, "Precise Ego-Motion Estimation with Millimeter-Wave Radar Under Diverse and Challenging Conditions," 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 2018, pp. 6045-6052, doi: 10.1109/ICRA.2018.8460687.
Chandran, Manjari, and Paul Newman. Motion Estimation from Map Quality Millimeter Wave Radar. https://www.robots.ox.ac.uk/~mobile/Papers/IROS06-Manjari.pdf.

ELECTRICAL & COMPUTER ENGINEERING
UNIVERSITY of WASHINGTON

ADVISORS: Prasad Shamain (Amazon), Mahmood Hameed (UW)
SPONSORS: Amazon