Boston University
Electrical & Computer Engineering
EC464: Final Test Report
Team 12: Hazard Harbingers

Team Members: Julia Hua, Noah Cherry, Timothy Borunov, Abdulaziz AlMailam

Equipment:

Hardware:

- Aerial LiDAR System:
    - DJI Inspire 1 Pro (drone)
    - SICK multiscan100 LiDAR
    - Jetson Orin Nano
- Embedded Nodes System:
    - SAMD21 Pro RF
    - RMF95W 915MHz Transceiver
    - BME688 Sensor
    - 128GB SD
    - I2C SD Card Breakout

Software:

- Shell script to process LiDAR data: processData.sh
    - FUSION/LDV library: GroundFilter, GridSurfaceCreate, CanopyModel, TreeSeg, DTM2XYZ, Cover
    - LAStools library: txt2las
    - density.bat script
- Python program: visualize2D.py
- ROS1 and Sick libraries installed with rviz on Linux device
- SOPAS ET software on windows device
- ROS package laser_assembler
- PDAL package

Setup:

The setup had three parts: LiDAR data acquisition, LiDAR data processing and visualization, and embedded nodes. For LiDAR data acquisition, the LiDAR and a Jetson Nano were mounted onto the drone using a 3d-printed mount. The LiDAR was connected to the Jetson Nano with Linux OS, ROS1, and SICK libraries. The drone can then be flown around trees with the LiDAR taking scans and the Jetson Nano saving the data into a ROSBAG file. Unfortunately, while we can collect data with this setup, we were not able to fly the drone yet due to safety concerns about the weight imbalance on the drone.

For the data processing and visualization, we took a scan with the LiDAR. The resulting ROSBAG file was subsequently converted to a single PCD file using laser_assembler and then to a LAS file using PDAL. To create the models, we require LiDAR data with trees, so this step was done with sample LiDAR data. The LAS file was inputted into the processData.sh script and produced multiple models: DTM, CHM, tree clusters, canopy cover, and vegetation density at various height stratas. The tree clusters, canopy covers, and vegetation densities can be better visualized as a 2D heat map.

For the embedded nodes we did a live demonstration simulating how the nodes would interact when deployed. Due to the space limitations of the testing environment, we did not attach antennas to prevent overloading the receiver and limit interference of local 900MHz networks. We also could not achieve enough distance to exit the transmission range, so to simulate entering a sensor node’s field we manually powered on the sensor. The collector constantly attempted to collect data via a broadcast message until the sensor woke up. The sensor then received the broadcast and dumped the SD card data back to the collector, which was then written to the SD.

Measurements:

LiDAR Data Acquisition: The LiDAR and Jetson Nano were successfully mounted onto the drone, and the LiDAR can successfully take and store a scan.

LiDAR Data Processing: The resulting ROSBAG file successfully assembles into a single PCD file and is then converted to a LAS file. Additionally, the models and 2D heat maps were successfully produced.

Embedded Nodes: The correct packet information in the terminal was successfully displayed when sent. The data was correctly copied to the collector SD card after transmission.

Conclusions:

The aerial LiDAR system has been built and is ready to acquire LiDAR data, and the LiDAR data can be successfully converted to a LAS format to produce the DTM, CHM, tree clusters, canopy cover, and vegetation density models. Both have been successfully tested in a lab environment and the drone is ready for further experimentation in a fly-safe setting.

Additionally, the embedded nodes correctly interact and are ready to be deployed. Sensor nodes collect data on the specified interval and store the information in CSV format on the SD card via SPI. When the collector node sends the wake message, the sensor node wakes, reads the SD card data, and transmits over LoRa to the collector. Correct transmission can be seen in the output we redirected to the serial terminal matching that of the SD card on the sensor node. The collector node then stores the data on its own SD card, and the sensor returns to sleep to prevent it from repeatedly sending data.

In the last few weeks, we will focus on flying the drone in a safe environment and acquiring a proper 3D point cloud. We also intend to fix the method for concatenating the point cloud data in a consistent and automated manner. Then, we can evaluate the models derived from the point cloud. We will also deploy three embedded nodes and create a live visualization for each of the sensor data. The embedded node hardware will also be moved to a more permanent perf board to prevent faulty wiring and compact the design. Given time we will also redesign the embedded enclosure to a more visible and compact design.