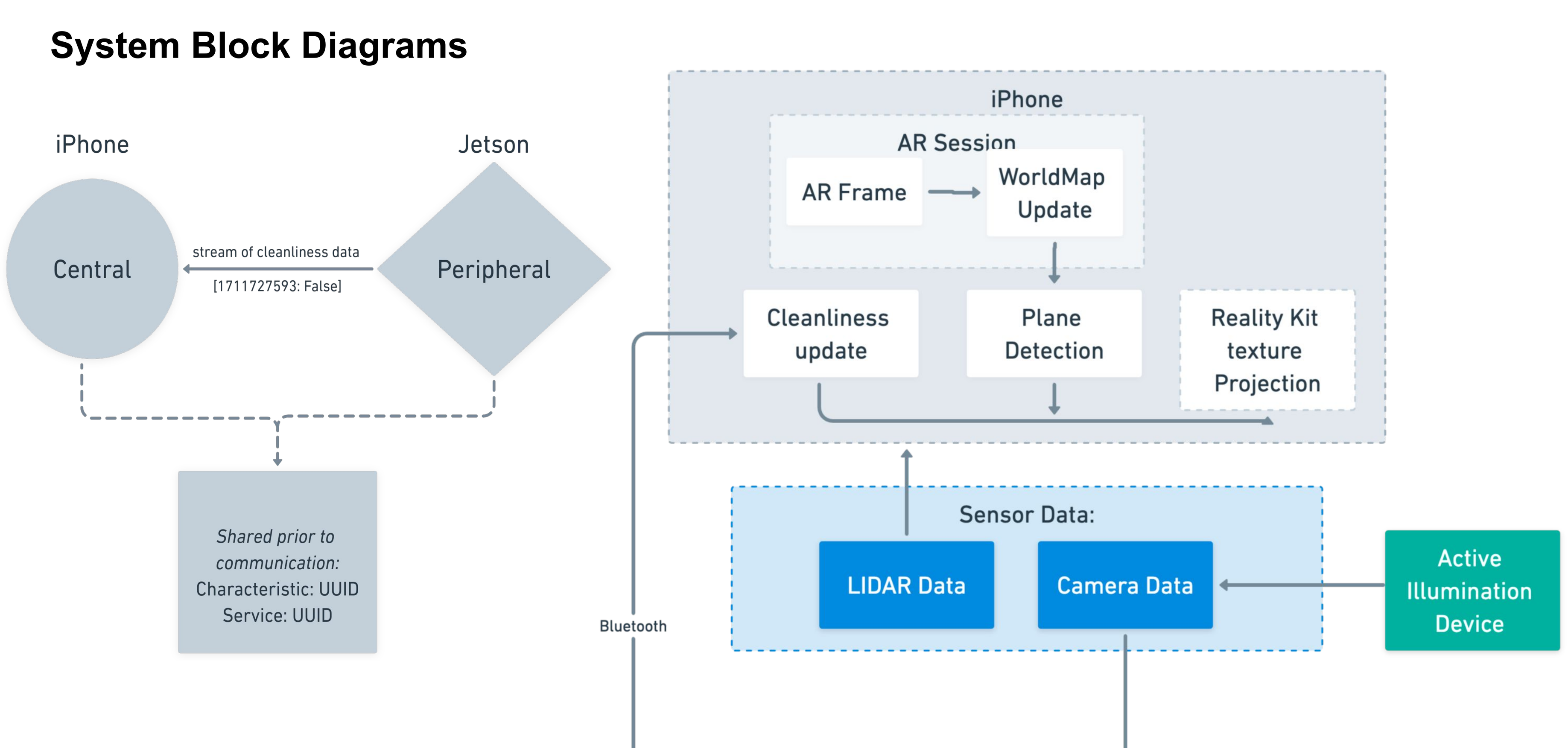
CleanAR

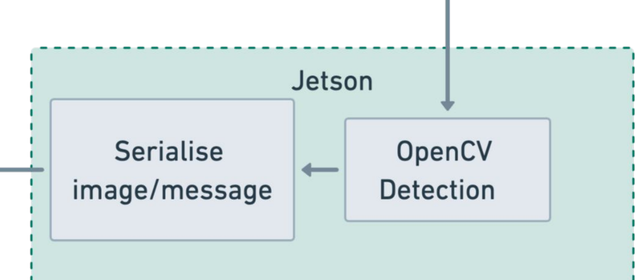
C1: Erin Gao, Nathalie Jeans, Harshul Singh
18-500 Capstone Design, Spring 2024
Electrical and Computer Engineering Department
Carnegie Mellon University

Product Pitch

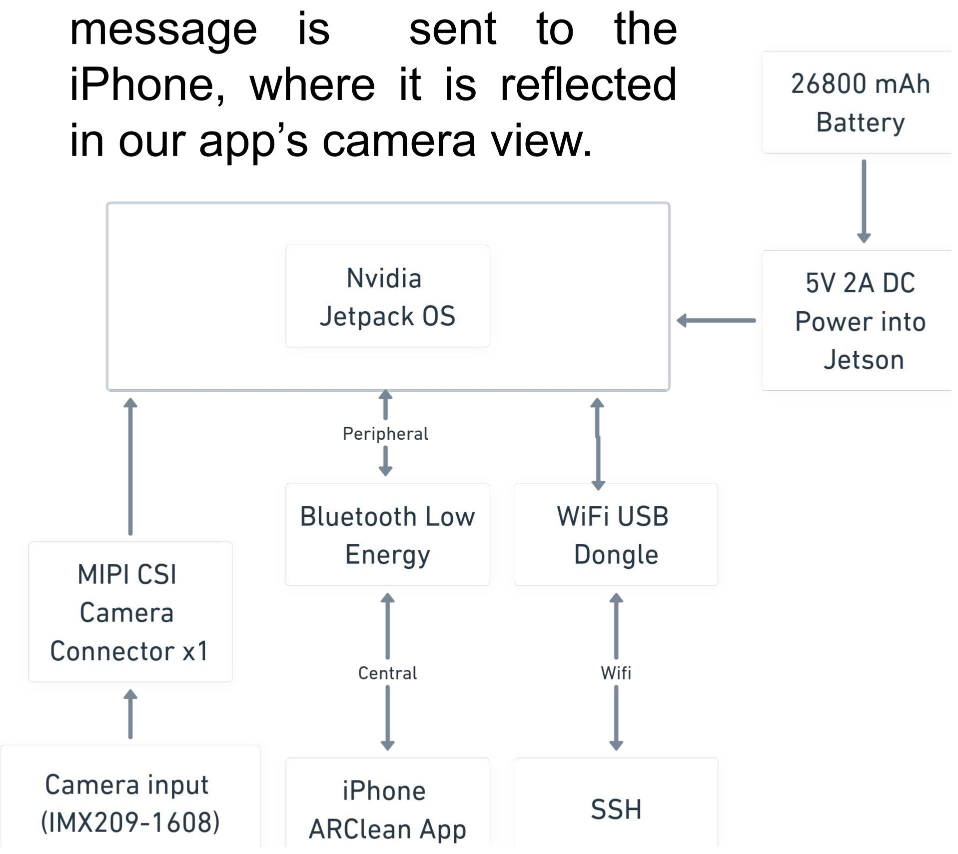
This system maps a floor with an Augmented Reality overlay to improve baseline vacuuming coverage and cleanliness. The application tracks uncovered floor space, erasing parts of the overlay that have been passed over by the vacuum. A camera behind the vacuum detects whether the floor is clean, highlighting areas where dirt is still present. This system is constrained to only work on white, patternless flooring, and the device should be able to run for four vacuum charge cycles. System requirements state that at least 90% of dirty flooring should be labeled as such, and that communication delay between the Jetson computing source and the AR application should not exceed six seconds. Test results confirm that both these use case requirements are met, with dirt detection achieving 93.75% accuracy and data transmission thresholded at 0.12s in testing.

System Architecture





Our system classifies and maps the boundaries of the floor using LiDAR sensors from the iPhone. In parallel, the Jetson camera performs dirt detection by using OpenCV to analyze the captured frame. A serialized message is sent to the iPhone, where it is reflected in our app's camera view.



Jetson I/O

Bluetooth Low Energy (BLE) communication sends data from Jetson to the iPhone. Jetson acts as peripheral (server) and iPhone acts as central (client) device. Data sent is a time stamp and boolean serialized as a string, denoting whether camera view captured dirt at the specified timestamp. '$' character is sent to denote EOM.

Our system requirements include criterion regarding the accuracy in initial mapping boundary detection to be ±10 cm. In addition, we require that at least 90% of dirty surfaces are marked as dirty. Dirt is defined as the existence of >1mm diameter particles.

System Description

Augmented Reality Mapping & Tracking

- Performing image detection on reference image provides fixed tracking position as the vacuum angle changes
- Plane detection and localization in the world through ARKit API
- Project line to track vacuum position along the floor plane
- Bounding Box Restrictions to avoid tracking vacuum outside floor boundaries

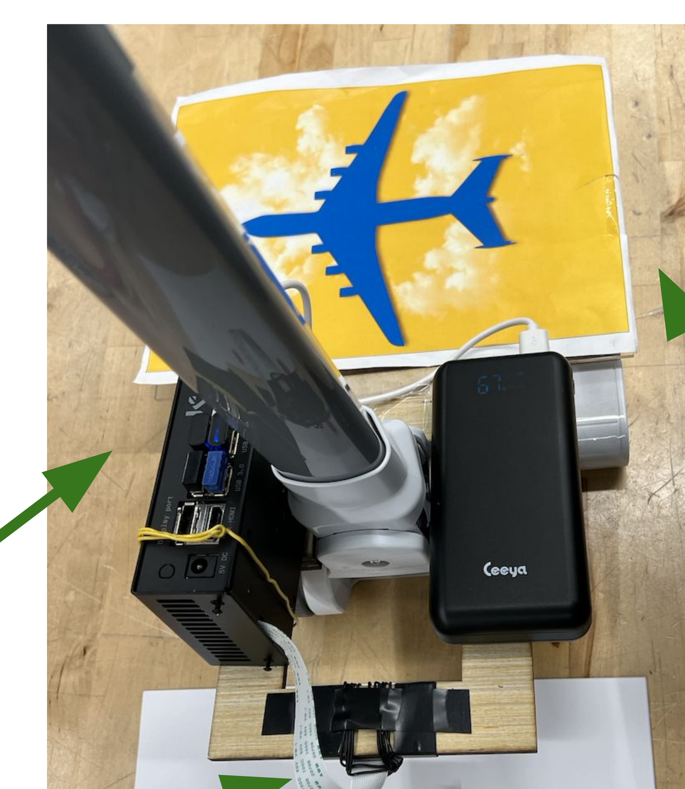
Bluetooth (Low Energy) Communication

- Apple restricts raw Bluetooth messages from Linux machines, therefore a pub/sub pattern is a more efficient protocol
- Jetson acts as a peripheral that broadcasts data to the iPhone app acting as a central
- Medium to transmit sensor data captured by the Jetson

Jetson Dirt Capturing & Processing

- Using contours to detect Dirt
- Sharpening image to detect dirt particles more clearly
- Bias towards false positives over false negatives
- active illumination inspired by dyson to generate shadows to enhance the visual footprint of dirt particulates

NVIDIA Jetson



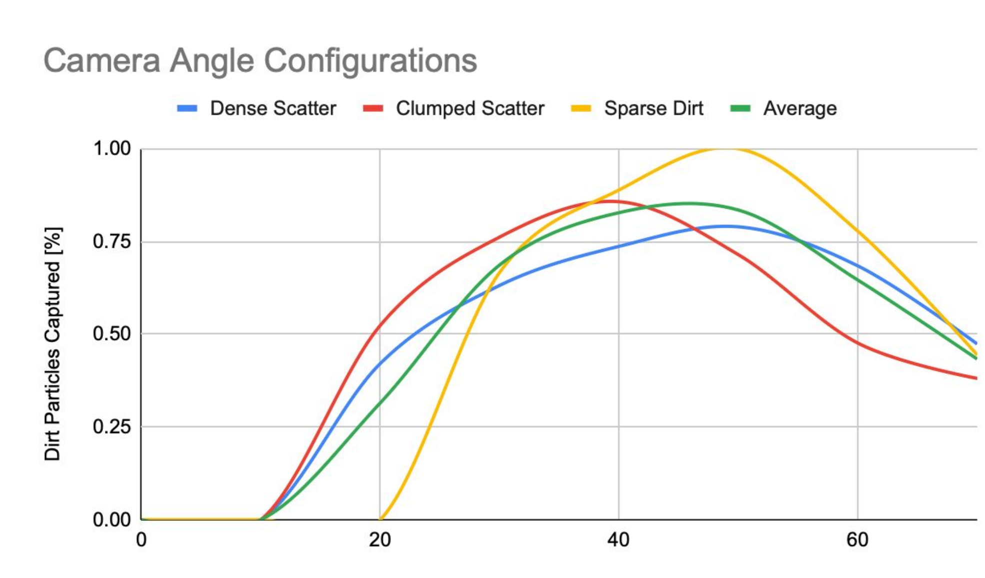
Orientating Image

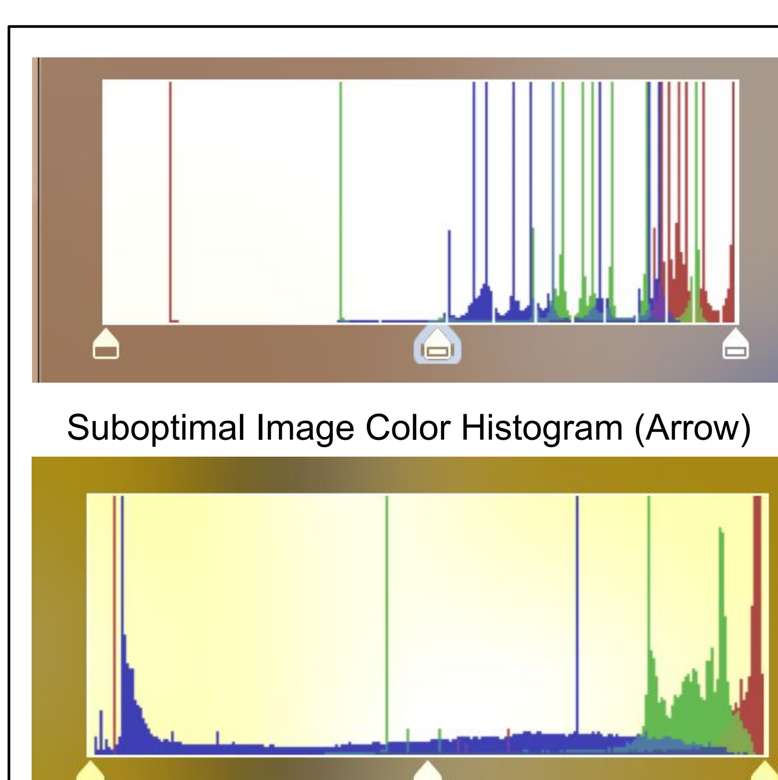
NVIDIA Jetson Camera

Sensing Module + Tracking image

System Evaluation

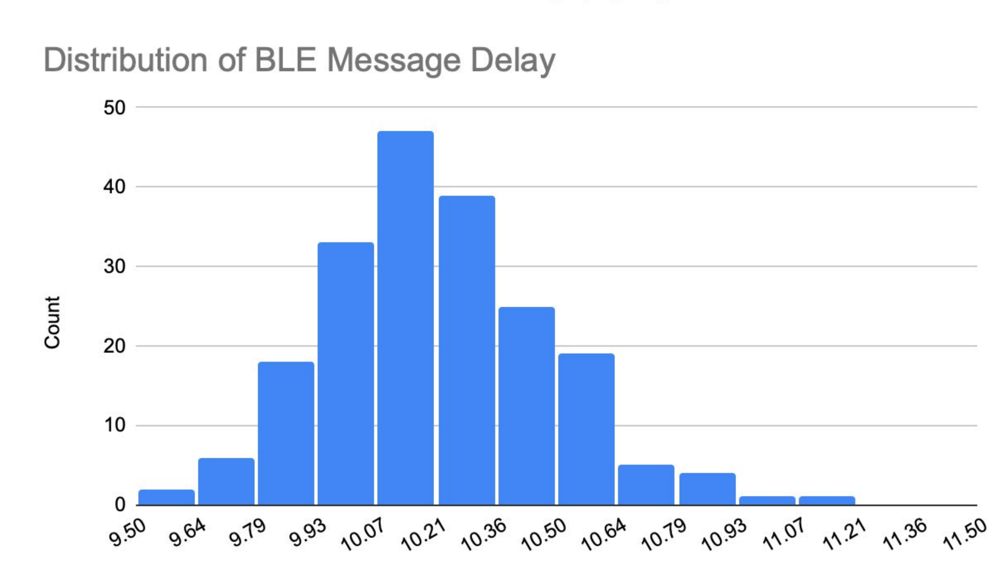
Multiple dirt configurations were set up for each suite of tests that we ran (clean, sparse dirt, dense dirt). Similarly, we ran tests to determine the optimal camera position and angle. We scoped performance of the AR application in multiple rooms in different settings: sunny, overhead lighting, dimly lit, dark. In each test case, the greatest distance between the mapped and true floor remained under 10 cm.





Suboptimal Image Color Histogram (Arrow)

Optimal Image Color Histogram (Airplane)



The above color histograms were used to inform reference detection images. Communication and mapping tests to determine the proper offset and pictured camera view as calibrated by the UNIX timestamp.

Conclusions & Additional Information

http://www.ece.cmu.edu/~ece500/projects/S24-teamc1

Lessons Learned:

- Apple implements many system constraints
- Conduct more research on different APIs and hardware before starting design
- Clearly define testing environment and requirements
- Sync with groupmates often
- Log all procedures for reproducibility

We spent more budget than anticipated and we had to tune some of the system conditions after the first couple iterations of our product. Fortunately, we were able to achieve all of our use-case requirements at the end.

Electrical & Computer ENGINEERING

Carnegie Mellon