Drone Based Measurement of Signal Propagation in Urban Environments and Analysis

Pranav Badami
Adviser: Kyle Jamieson

Abstract

ABSTRACT PLACEHOLDER

1. Introduction (In Progress)

### 1.1. The Problem: ISP Monopolies in the United States

Home Internet users in the United States are at the mercy of their Internet Service Provider(ISP). Users seeking relatively modern download (> 25 Mbps) and upload (> 3 Mbps) speeds for home Internet are extremely restricted in their choices. The FCC states that 30% of Americans (measured in developed census blocks) have no ISPs delivering home Internet at these speeds, while another 48% of Americans have only one ISP providing service at these speeds[**?**].

As a result of this lack of choice (and competition), American home Internet are forced to pay disproportionately high prices for relatively slow connections. For example, Americans pay higher prices for home Internet than Europeans across all categories of broadband speed[cite connectivity].

The plight of the American home Internet user is exposed even more when considering the highest available home Internet speeds for affordable Internet plans. While residents of Seoul and Hong Kong can choose among plans offering 300 Mbps connections in the price range of $35-$50, no American city has any plans offering > 50 Mbps speeds for the same price range. [cite connectivity] The much cheaper and faster connections available in Asia demonstrate that it is technologically feasible to provide fast home Internet connections for a low price.

Not coincidentally, ISPs in Asia operate in a much more competitive landscape, forcing them to provide quality service at a lower price. In South Korea, for example, upstart ISPs can lease, and

Figure 1: A comparison of prices for various categories of broadband speeds in the United States versus Europe.
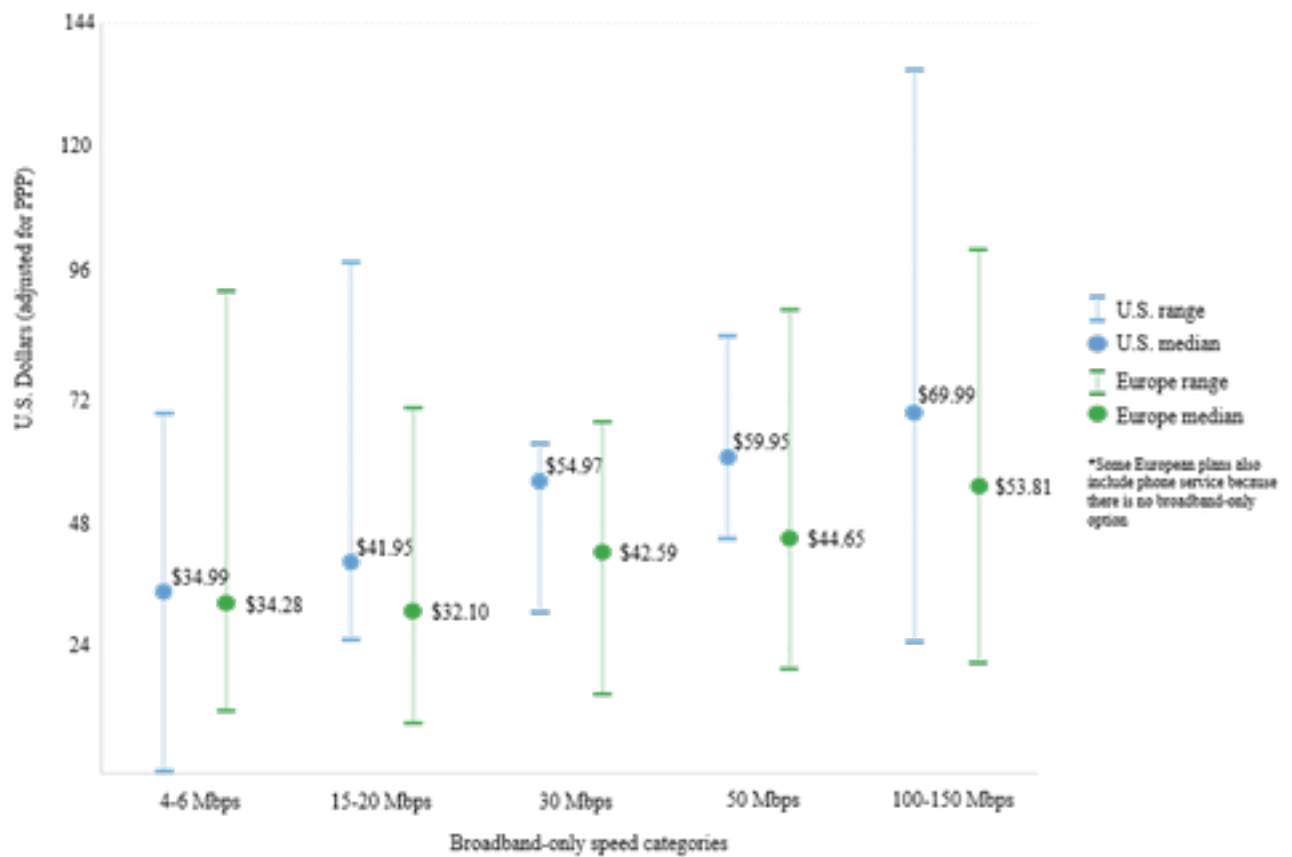
sell, unused bandwidth from a public utility that reaches the vast majority of the clustered, urban population. Further, the South Korean government forced Korea Telecom to open its network in the '90s, leading to increased competition and lower prices for users[cite wired].

Further, the population of the United States is large and spread out over a vast expanse of land. Reaching more homes in the United States requires laying lots of costly wire, which is often prohibitively expensive. Consider the efforts of Google Fiber. Google Fiber aims to offer connection speeds of 1000 Mbps at a comparable price to existing plans, reaching consumers in select neighborhoods of select American cities[cite google fiber]. According to Google, the process of bringing Fiber to a city involves initial exploration, network design, construction (laying fiber), and finally, customer sign up and installation. A Goldman Sachs report on Fiber, estimated that it would cost around $140 billion for Google to cover the whole country[cite Goldman]. Further, the

involved process required to dig and install fiber is extremely time consuming. The cost and time required to lay fiber across a wide geographic area makes it extremely difficult for even established companies like Google to enter the ISP realm, much less a host of smaller companies or startups. . In 2014, Tom Wheeler, Chairman of the FCC, acknowledged the lack of competition among ISPs and recognized a need for the FCC to do more to protect and create competition in the United States [cite Wheeler].

1.2. A (Potential) Solution: Wireless Mesh Networks

With the cost of establishing novel, last-mile wired networks being so high, there has been a lot of work on establishing wireless networks. One such notable project is Roofnet, a wireless mesh featuring over 40 nodes spread out over the urban area of Cambridge, Massachusetts. Roofnet nodes consist of a small PC, an 802.11b card, and an 8 dBi omni-directional antenna. These nodes are primarily installed on three to four story tall buildings, with packets hopping wireless nodes until they reach a wireless node which is also connected to a gateway to the rest of the wired Internet[cite roofnet]. Individual users install these nodes on their roofs or windows and join the network.

Similar to Roofnet, another community powered mesh exists in Red Hook, Brooklyn[cite NYT Red Hook]. The goal of these networks is to provide a cheaper, more accessible form of Internet architecture in comparison to the extremely expensive deployment of fiber cables. However, these wireless networks face challenges operating in the urban environment.

1.3. Challenges with Wireless: Propagation and Measurement

The propagation of wireless signals in a noisy, obstacle-ridden urban environment often results in wireless networks operating below peak efficiency. Buildings and other environmental obstacles, such as trees, can greatly increase the path loss of a wireless link between two nodes. Then, establishing a line-of-sight (LOS) connection between two wireless links is important for low-loss links and robust wireless networks. However, the Fresnel Zone (an ellipsoid-shaped volume) around the LOS must also be clear of obstacles for optimally low path loss between a transmitter and receiver.

3

Much work has been conducted on measurement campaigns of wireless signal propagation. Theodore S. Rapaport has conducted many such campaigns in a variety of settings in order to explore real-world propagation in different environments. For example, Rapaport, Durgin, and Xu's "Measurements and Models for Radio Path Loss and Penetration Loss In and Around Homes and Trees at 5.8GHz" conducted such a campaign in a suburban area. Signal strength measurement were taken by moving a receiver around a 1m square area and averaged to represent the path loss for a given room or outdoor location[cite Rappaport]. Such campaigns, while useful for sampling signal strength at key locations, don't give a complete understanding of signal strength over a space.

Other campaigns have attempted to get a more complete measurement in an urban environment. A group of Brazilian researchers conducted a measurement campaign in Rio De Janeiro with a fixed transmitter on the window of an 8 story building. They built a receiver block with GPS and collected measurements by driving around streets surrounding the building housing the transmitter. GPS and signal strength measurements collected by the receiver block are joined to construct a map of measured signal strength at street level. While this may be useful for mobile Internet users at the street level, participants in a wireless mesh for home Internet may have receiver nodes at various heights, necessitating a more 3-D measurement of signal propagation in urban areas. Manual campaigns such as these use labor intensive techniques commonly referred to as "war walking" or "war driving"; these campaigns also have to be limited in scope due to the physical limitations of where measurements can be performed manually.

Other measurement campaigns have attempted to measure 3-D signal propagations using drone technology. Drones offer several unique features that can make a measurement campaign more robust and complete than traditional static or manually measured campaigns. DroneSense was an automated 3-D wireless signal measurement campaign created by a team of researchers from Dartmouth. The DroneSense system autonomously traversed indoor spaces using visual reference markers which trace a path through a space. These reference markers are physically placed in a space and the drone uses its on-board camera and computer vision techniques to identify markers and use these markers to autonomously navigate through hallways and rooms. The drone's onboard

wireless radio is used to measure signal strength.

Figure 2: Left: Continuous 2-D map of signal strength measurements in the streets of Rio de Janeiro (Authors). Right: DroneSense system navigating indoor space using reference points.

The DroneSense system can only measure signal strength in one frequency band because they rely on the drone's onboard wireless chipset; this doesn't allow the system to measure propagation under varying transmitter configurations. Regarding it's navigation module, DroneSense is limited in that it's navigation is reliant on physical reference points strategically arranged in 3-D space. This measurement technique is restricting in multiple ways. First, physical markers have to be created and laid out in 3-D space. While DroneSense is autonomous in terms of the drone's navigation, laying out the actual markers for the drone to navigate is still manual and limited by where markers can be placed. DroneSense also analyzes signal strength in terms of which reference point measurements were performed at. The reference points are points in 3-D space but do not have inherent spatial information encoded in them. Then, in addition to the manual placement of the reference points, DroneSense requires manual mapping of the reference points to 3-D space in order to create a continuous map of signal propagation. Lastly, DroneSense is limited to indoor measurement, where the measurement environment is much more contained and well behaved.

1.4. This Project: A More Robust Measurement Approach

Common measurement techniques such as "war walking" and "war driving" don't provide robust, complete data that allow us to understand real-world signal propagation in complex urban environments. Newer approaches using drone technology, while promising due to spatially versatile,

autonomous measurement, have also been limited in a variety of ways. Then, the question still remains: *how can we gather more robust data to better inform the design of wireless mesh networks*?

Primarily, the data should help us form a complete picture of how a wireless signal from a transmitter propagates through a given environment. A crucial element of this is to perform measurements of signal strength in the complete 3-dimensional space. This is important because in a wireless mesh network, users may place transmitters and receivers in many different spatial configurations; for example, end users may live or work on different floors of apartment buildings and place antennas at varying heights. While DroneSense attempted to perform measurements in 3-D, their usage of reference points meant that their measurements were limited by the spatial arrangement of these points and the path the drone took to get to each of these points. Instead, a robust data set that would allow us to better understand real-world propagation would be a 3-D map of propagation through an environment.

Mesh networks can also be developed in many different environments. Dense urban environments with taller buildings may necessitate a propagation map with emphasis on the vertical space between buildings. More spread out suburban environments with shorter buildings may require require measurement over large swaths of land. A robust measurement technique should be easily configurable to perform measurements in a variety of different environmental configurations. This versatile method would depart from the conventional methods of war walking and war driving; there are many environmental configurations where it would be extremely demanding or impossible for humans to be physically involved in the measurement process. For the measurement technique to operate without direct human input, it would likely have to navigate autonomously. Drones are growing increasingly technologically capable and accessible; this technology could be strategically used to perform truly 3-D measurements autonomously.

A robust 3-D data set of signal strength measurements could help inform best practices for mesh network design. Experimental methodology is a multi-faceted implementation choice that can greatly impact how insightful a signal propagation map can be towards real-world network configuration. Antennas in a mesh network can be placed in many different physical locations and

operate with a variety of configurations. For example, transmitters can operate in different frequency bands depending on their circumstances. In urban environments, a signal in the 2.4GHz ISM band can provide better penetration through obstacles than a signal in the 5 GHz band; however, certain crowded channels in the 2.4 GHz may cause undesirable interference and require transmission in a less crowded channel. Experiments should account for these varying configurations and the measurement technique should also allow for measurement across commonly used ISM frequency bands.

Lastly, there needs to be a digestible way to analyze the 3-D maps of signal strength across various experiments to determine which configurations worked better than others. This will require a combination of data visualization and data analysis. Visualizing the 3-D map can provide an intuitive understanding of how certain environmental features affect signal propagation in various configurations. Smoothing and aggregating the measured values over area or volume can provide metrics which help us digest the data and inform actual mesh network design.

1.5. PROJECT ACRONYM: PROJECT TITLE

Motivated by the desire to build more performant wireless mesh networks in cities and suburbs, [PROJECT ACRONYM] aims to develop a robust measurement technique to measure signal propagation in urban environments. [PROJECT ACRONYM] will demonstrate the ability to create a 3-dimensional map of signal propagation in an urban environment; further, [PROJECT ACRONYM] will perform measurement experiments in different environments with various configurations to show how the versatility and extensibility of this measurement technique. Lastly, an analysis framework will be developed to show how the 3-dimensional maps of signal strength, in concert with various experimental setups, can actually guide real world mesh network design.

To do this, we will build a signal measurement apparatus and fly it with a drone to conduct a measurement campaign in a handful of locations on the Princeton campus. A measurement apparatus collecting and storing signal strength measurements across the 2.4 GHz ISM band will be mounted on a drone. We will fly a drone on autonomous missions, where the drone will create

accurate, high-resolution internal flight data logs; over the course of a flight, these flight logs will document the drone's latitude and longitude coordinates with 1 meter precision, along with the drone's height above ground. We will join the drone's flight log data with the measured signal strength data recorded on our measurement apparatus. This joined data set will let us create 3-dimensional point maps of signal strength at frequencies across the 2.4 GHz band. The collected data can then be aggregated to get a comprehensive, robust 3-dimensional map of signal strengths around a certain urban feature, such as a courtyard or a street with buildings on both sides.

There are two overarching research goals of this project. The first is to successfully generate a 3-D point cloud with associated signal strength measurements in an urban environment. This data set will serve as the foundation for the subsequent research goals of the project. Experiments will be ran with various transmitter locations and configurations, including varying center frequencies and channel widths; a separate 3-D point cloud generated for each experiment. These sets of experiments will be performed at a handful of locations across campus. Then, another important research goal of the project is to be able to spatially aggregate these point cloud data sets into "buckets", where each bucket is assigned a value equal to the average signal strength of point measurements in that bucket. This would serve multiple purposes, making the data more intuitively understood and easily visualized. Also, this would allow us to normalize the data per unit of volume; normalized values can help compare the effect of certain transmitter configurations and environmental features on propagation.

Figure 3: A mockup of the main research goals of [ACRONYM]. Left: A 3-D point cloud of signal strength measurements, given a fixed position transmitter. Right: After analysis and aggregation, spatial buckets of average signal strength values are formed; this normalized data can be compared across experiments with various configurations.
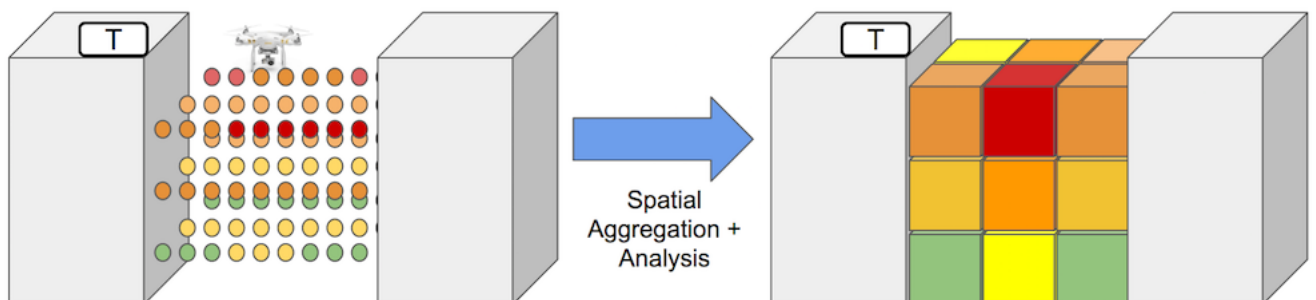
This approach is novel because it would yield a robust, 3-dimensional map of signal propagation; existing measurement techniques are restricted to 2-dimensional maps or static point measurements. Additionally, using a drone to fly autonomous missions and collect measurements makes this approach viable in many different environments where war walking or war driving would be impossible. Since this project is motivated by wireless mesh networks,conducting measurements at various heights in challenging urban spaces is important as users can be situated at various floors. The analysis framework we propose allows us to normalize and compare these complete 3-D data sets across many configurations to determine how best to layout a mesh network in an environment with certain features.

The remainder of this work discusses how the technological stack required by this approach was designed and built, and then proceeds to detail the methods of data collection and experimentation. Finally, it attempts to show how the proposed spatial aggregation analysis can inform better mesh network design in the future.

2. Implementation: Point Cloud of Signal Strength Measurements

Several implementation challenges presented themselves as we began to work towards the primary research goal. The following sections describe how we built the various components required to construct a 3-D point cloud of signal strength measurements. Mainly, transmitter configuration and receiver apparatus design are important design choices for any measurement campaign. The receiver apparatus had to measure and record signal strength over time and, importantly, be mountable on the drone. The drone then had to safely and autonomously fly with the measurement apparatus attached. The spatial data recorded by the drone had to be joined with the data collected by the receiver apparatus to generate the 3-D point cloud.

2.1. Part 1: Transmitting a Wireless Signal

2.1.1. Transmitter Selection Since this project was focused on measuring the performance of wireless networks in urban environments, it was imperative to choose a transmitter which would be typically used in an urban setting. Consider a mesh network configuration in a town such as

Princeton. This mesh network would need to meet several key criteria. First, it would need to have a way to connect to the wired Internet at some wired gateway, which could be many miles away or even in another town altogether. One reasonable way to establish this connection would be with the use of high-throughput, carrier-grade, long-range backhaul links which can establish wireless connections at ranges upwards of 100 km [Ubiquiti cite].

End users of the mesh network living in Princeton would not be connecting directly to these backhaul links, however. Backhaul links would need to be placed at high elevations and meticulously aligned with one another to establish a LOS connection. It may be impossible for users to establish LOS connections with backhaul links because users will often have windows or roofs with obstructed views; further, they will also have pyhsically smaller, lower power antennas where they may not be able to transmit to a distant backhaul link.

Instead, a third kind of antenna is required to fill in the gaps between user bridge antennas and powerful backhaul links. A Point-to-Multi-Point (PtMP) link, which transmits to several receiving antennas, is required to complete the mesh network configuration. These PtMP links are more cost effective and smaller than backhaul links; they can be placed around the town of Princeton and serve as connection points for the end users who live on the same street, housing development, or intersection as one of these links.

For this project, the Ubiquiti NanoStation M2 will be used; the NSM2 is a PtMP link which operates in the 2.4GHz spectrum. The NSM2 is appropriate for an urban mesh network because it has good 180-degree coverage in the azimuth and elevation planes. Then, if it is placed on top of a building facing the street, it can ideally provide a strong signal to a number of other buildings on the same street at a variety of elevations (floors); a large number of users would be able to connect to the locoM2 from their windows and join the wireless mesh.

2.1.2. Transmitter Selection Ubiquiti transmitters also allow for fast and easy configuration through the built-in airOS operating system. The airOS interface is accessed through a web browser and allows the user to change the operating mode of the device, the channel width, and the center frequency of the device; these parameters will be varied during the experimentation phase. For this

Figure 4: Radiation patterns of the Ubiquiti NanoStation M2
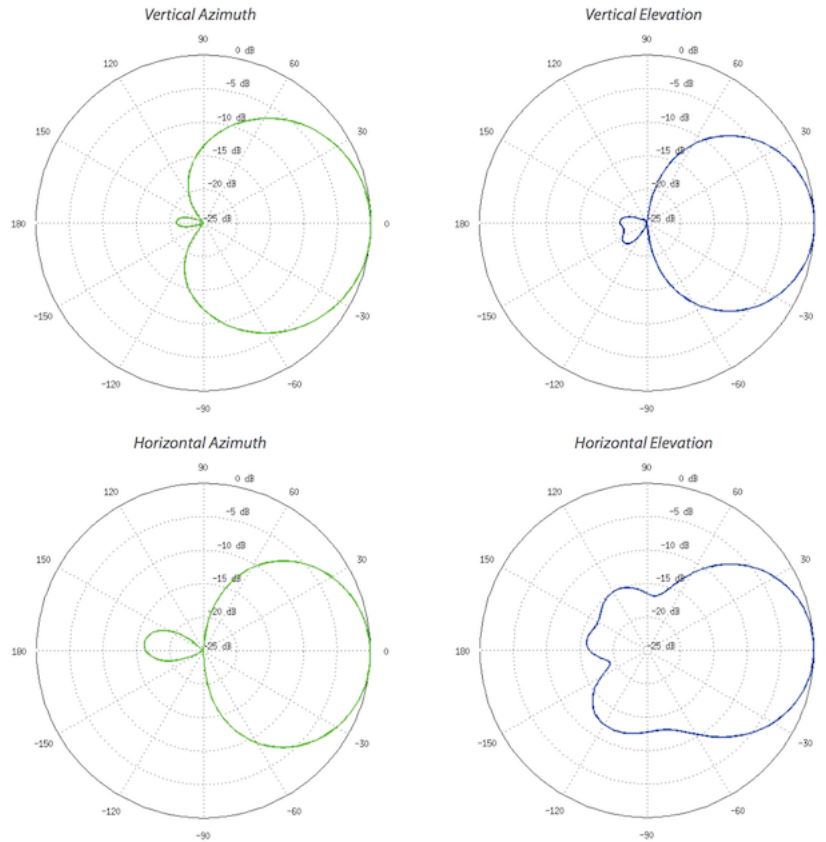
implementation the Ubiquiti transmitter will receive power-over-ethernet (PoE) from a portable power supply and PoE adapter. The transmitter will also be mounted on a telescoping tripod, which will allow us to experiment with the transmitter at various heights if it's not possible to move between floors.

2.2. Part 2: Measuring Wireless Signals in 3-D

In order to create a 3-D point cloud of signal strength measurements, measured values need to be combined with 3-D spatial data. Two major implementation components are needed to achieve this goal. First, a drone is required to collect spatial data across various planes at many different heights. A signal measurement apparatus which can detect signals across the 2.4GHz spectrum and then record these measurements is also needed. If this apparatus can somehow be mounted on the drone, we could feasibly gather wireless signal data throughout a 3-D space. The data collected on these separate devices can then be joined into a single data set representing the 3-D point cloud of signal

strength measurements.

2.2.1. 3-Dimensional Data: Collected via Drone Collecting precise and accurate spatial data would only be possible if a reliable drone with state-of-the-art battery life could both log flight data and be autonomously controlled. Autonomous control is imperative to fly the drone in specific flight patterns, which can be overlayed on each other to produce the robust, complete dataset required for the 3-D point cloud. The DJI Phantom 3 Advanced is a "pro-sumer" drone which produces incredibly precise spatial data using multiple built-in stability and guidance systems. DJI is a world-leader in producing drones for hobbyists as well as professionals. While other companies such as Parrot Drones sell devices with open-source hardware and software for increased customizability, this project used a DJI drone which was less customizable but more reliable. Reliability and fault-tolerance was a key consideration because drone flights would be conducted in urban settings more prone to signal interference where drone failure could cause harm to property or people. Further, because of DJI's large market share, many third party apps leverage the extra features of the Phantom 3 Advanced that would potentially have to be created separately if another brand was used.

Battery life was an important consideration because many autonomous flights would have to be conducted in a given space, each lasting several minutes. With an approximate 23 minutes of flight time out of the box, the P3A offers one of the longest flight times available in the consumer drone space. The P3A was used instead of the Phantom 4, a more recent model, because it has has a removable camera gimbal which was leveraged to build a mounting bracket housing the measurement apparatus (see Measurement Apparatus section, Mounting section).

2.2.2. Spatial Data Collection The P3A features an optical stabilization system for micro-level positional adjustments and a robust satellite positioning system for macro-level navigation purposes. DJI's "Vision Positioning System" (VPS) features two ultrasonic sensors and a monocular camera on the bottom-side of the drone.These sensors are always-on and the P3A uses the ultrasonic devices to measure distance from the ground and the camera to identify patterns under the drone; this input can be used to stabilize the drone such that it can hover in place or closely follow a pre-defined path. On the other hand, the P3A's satellite positioning system uses both GPS and

GLONASS (the Russian satellite network) to provide geographic location services. The inclusion of GLONASS effectively doubles the number of satellites the P3A can fix onto; throughout testing and experimentation, the P3A was regularly fixed with 14+ satellites, providing robust, 1 meter accuracy. The geographic coordinates and the height measurements gathered by the satellite position system and VPS, respectively, are logged on automatically created flight logs and also transmitted to the drone's controller in real-time.

2.2.3. Precision and Automation: Litchi iOS app Many third-party apps take advantage of these systems to provide autonomous navigation on the P3A. Litchi is an app which features a "Waypoint" mode where users can construct a path out of waypoints for the drone to follow. Custom heights and speeds can also be set for the waypoints; the drone's heading can also be manually controlled along the route. Specifically, Litchi allows the user to control the drone's heading using "points of interest"; for this project, the location of the transmitter will be a POI the drone will orient towards. At the start of a waypoint mission, the entire path is wirelessly uploaded to the drone which performs an auto-takeoff maneuver to begin the autonomous mission. The drone travels from it's starting location to the first waypoint and continues to follow the path until it reaches the last waypoint.

The Litchi app also integrates with AirData UAV to automatically sync flight logs after each completed flight. AirData UAV serves as a repository that allows users to download flight logs much more easily than accessing the internally stored flight logs on the P3A. Data can be downloaded in CSV format for ease of analysis, or KML for ease of integration with mapping services such as Google Earth. Logs include timestamped information collected from the drone's built-in systems, include the machine's height, latitude and longitude. Geographic data is reported in decimal notation, while timestamps are in UTC-3 format.Litchi flight logs also contain fields detailing how long the machine has been flying (in milliseconds) as well as the flight state; since missions occur autonomously, the drone takes-off and lands automatically and records these maneuvers as distinct flight states. When the drone is traversing between waypoints, it logs a dedicated waypoint fly state, allowing for a convenient way to filter data when constructing the point cloud.

Figure 5: Left: Waypoint mode in the Litchi app, viewing a mission. Purple markers denote the waypoints, which the drone will traverse in numbered order. The clouds above each marker indicate the height the drone should assume at a given waypoint. The opaque blue paper airplanes under the waypoints indicate the drone's heading. The blue marker is a point of interest. Right: Path followed by the drone during a test of this waypoint mission; the drone took from the H (home) marker. Note that waypoints are separated by < 16 ft, demonstrating the granular position control of Waypoint mode.

Figure 6: An example flight log from the Litchi app, automatically synced to and downloaded from the AirData UAV repository. Note the change in the "flycState" column as the drone transitions from taking-off to executing a mission. (This is only a small subset of columns from the flight log.)

	latitude	longitude	ascent	flight_time	datetime	satellites	flycState
9	40.282239	-74.585949	2.624672	900	2017-04-17 17:49:42	15	AutoTakeoff
10	40.282239	-74.585950	2.624672	990	2017-04-17 17:49:42	15	AutoTakeoff
11	40.282239	-74.585950	2.624672	1090	2017-04-17 17:49:42	15	Waypoint
12	40.282238	-74.585950	3.280840	1290	2017-04-17 17:49:42	14	Waypoint

2.2.4. Drone Safety: Features and Precautions Since we would be conducting drone flights on the Princeton campus, we had to be certain that the drone would take appropriate fail-safe measures in case something went wrong during an experiment. For example, the connection between the drone controller and the aircraft is a wireless connection in the 2.4 GHz band; what would happen if the connection between controller and aircraft was broken due to heavy interference? Or simply if the controller battery died?

First, we tested how the drone would perform if it lost connection to the flight controller during a manual flight. The drone's behavior in this scenario can be configured in the aircraft settings menu through both the DJI GO or Litchi apps. Since the drone would not be flying directly

14

above buildings, obstacles, or people during our experiments, it was configured to perform and auto-land maneuver if connection with the controller was lost. To simulate losing connection with the controller, we simply turned the controller off while the drone was in flight. This was tested 3 times and the drone landed successfully each time. Then, we also wanted to test how the drone would perform if it lost connection with the controller during an automated waypoint flight. This behavior can be set in the mission settings menu in the Litchi app. Since the drone auto-lands once it reaches the last waypoint of the mission, we opted to configure it to auto-complete the mission even if it lost connection with the controller. We tested this 3 times and the drone successfully completed it's mission each time.

When the drone is performing automated flights in waypoint mode, the controller is set to Function, or "F", mode; manually controlled flights occur when the controller is in Positioning, or "P", mode. Another potentially unsafe scenario could be when the drone malfunctions (possibly due to a compass or VPS error) while the controller is in "F" mode and manual control is disabled. We wanted to ensure that switching the controller to "P" mode would successfully restore manual control of the aircraft in case it needed to be steered out of danger and landed. Switching modes and landing the craft was tested successfully three times as well.

2.3. Receiver Apparatus Overview

The design of the receiver apparatus was heavily constrained by weight. In total, the receiver apparatus weighted (TODO: X). If we were to mount the apparatus on a drone, it would certainly need to be light enough for the drone to fly and be controlled reliably. Another important constraint was the ability to interface with the equipment, since the collected data needed to be logged for later analysis.

At a high level, the receiver apparatus would need to measure and log signal strength across the 2.4 GHz band. Further, it would also need to log timestamps for these measurements; this is because the spatial data on the drone would need to be joined with the spectral data from the apparatus to generate the desired point cloud of signal strength. Aggregating and comparing this point cloud

data could inform best practices for transmitter and/or receiver placement in various urban settings.

The receiver apparatus was built on a light wood board and contained a Raspberry Pi, spectrum analyzer, and a portable battery pack; a measurement and logging script ran on the Raspberry Pi to collect and store data. The following sections describe each of these design elements in more detail.

Figure 7: Layout of the receiver apparatus, which is to be mounted on a DJI Phantom 3 drone.
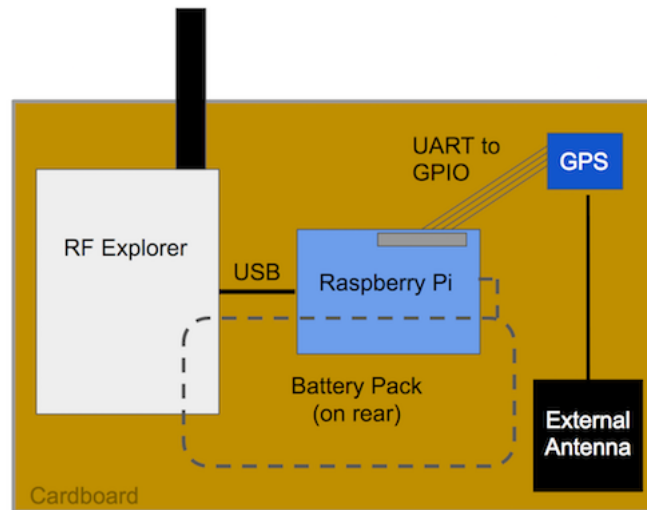
2.4. On-board Computing and Logging

A Raspberry Pi 2 Model B served as the on-board computer of the receiver apparatus. The Raspberry Pi was selected for it's versatile interfacing options and expandable flash storage, as well as it's very light weight (TODO: weight), power efficient design (200mA operating current). Specifically, the Model B has 4 built-in USB ports as well as 40 General Purpose Input/Output (GPIO) pins. The Pi ran a Raspbian Jessie operating system mounted on a 64 GB SD card, which stored the collected data. To power the Raspberry Pi, a lightweight (X g) lipstick battery with a capacity of 3350 mAh was used; this meant that the Pi could be powered for many hours at a time, which was a crucial feature later required to perform experiments (see Generating a Point Cloud: Setup section).

2.5. Spectrum Analyzer

The receiver apparatus needed to have an actual receiver to measure the strength of wireless signals in the 2.4GHz. This project used the RF Explorer handheld spectrum analyzer, a relatively

lightweight device (185g) that can run off its own internal battery and features a mini USB port for interfaceability[cite Seeedstudio]. RF Explorer allows for easy configuration of minimum and maximum frequencies detected as well.

To collect data from the spectrum analyzer, the RF Explorer was connected to a Raspberry Pi(see "Logging and Compute") with a serial connection over USB. The RF Explorer sends bits over the serial connection; these bits are structured as "sweeps" of signal measurement data. On each sweep, RF Explorer sends the measured signal strength for frequencies from its configured minimum to maximum frequency; in this project, the lower frequency bound was set to [X] MHz and the upper bound was set to [Y] MHz.

RF-Explorer-For-Python (RFEP) is a Python library with an API for detecting, connecting to, and reading from an RF Explorer device. Upon instantiating a RFECommunicator object provided by RFEP, the user can connect to the RF Explorer by configuring the RFECommunicator connection as a serial connection with the appropriate USB device port. The RFECommunicator then provides utility functions to read sweeps of data, returned as arrays of length [TODO: X] with frequency steps of [TODO: Y] MHz. This functionality will allow us to simultaneously gather and log signal strength data across the transmitting frequencies of our PtMP link, as determined by the transmitter's configured center frequency and channel width.

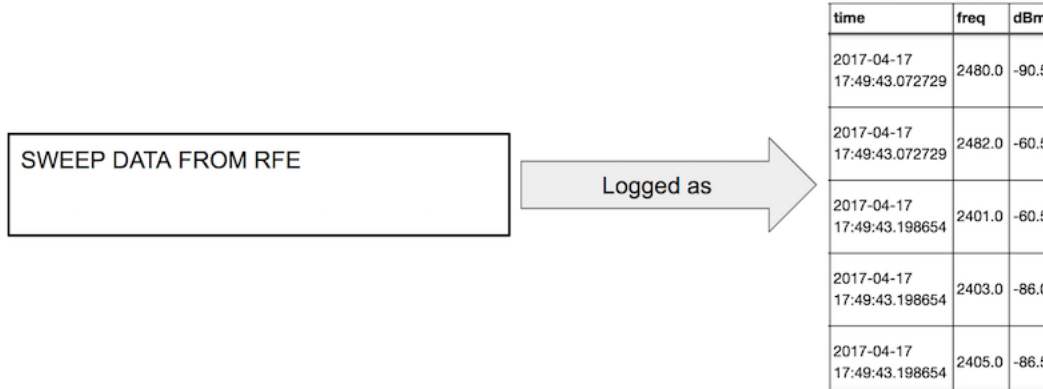## 2.6. Measurement and Logging Script

In order to construct the 3-Dimensional map of signal propagation, the receiver apparatus would need to gather data from the spectrum analyzer and log this data in the Raspberry Pi's onboard filesystem. As mentioned earlier, the spectrum analyzer communicates with the Pi over serial connections.

First, data from the RF Explorer was collected at 50000 baud via the USB serial connection using the RFECommunicator object from the RFEP library. The RFECommunicator object featured several convenient utility methods to get parsed signal strength data in an array. The measurement script initially connected to the RFE and configured it as necessary for any given experiment that

17

had to be performed. After a serial connection over USB was established, the RFE measurement configuration was read, allowing us to fix the minimum frequency and the step frequency. The first value in an array of sweep data corresponded to the signal strength at the min frequency while the value at some index i corresponded to frequency $min_f req + i * step_f req$. After setting these configuration variables, the measurement script entered a `while` loop where it continuously read data from the spectrum analyzer as new sweeps of data became available.

Frequency and signal strength were logged as timestamped rows (using python's `datetime`) in a file in the Raspberry Pi's filesystem. To increase the utility of the data collected on the measurement apparatus, the Pi logged measured signal strength at every frequency step in every sweep gathered by the RFE. The RFE had an update rate of around 5 Hz; a test run counting the number of RFE sweeps in 100 seconds returned a tally of [X], including the few seconds required for the RFE to set up and begin sampling its configured frequency band. Based on this update rate, the measurement script logged approximately 1 MB of data per minute of runtime.

Figure 8: Raw sweep data from the RFE was read by the Pi in a while loop and logged as rows in a data file. Note that RFE data is not timestamped; timestamps were obtained using `datetime` in Python after the Raspberry Pi clock was aligned with `ntp`.

time	freq	dBm
2017-04-17 17:49:43.072729	2480.0	-90.5
2017-04-17 17:49:43.072729	2482.0	-60.5
2017-04-17 17:49:43.198654	2401.0	-60.5
2017-04-17 17:49:43.198654	2403.0	-86.0
2017-04-17 17:49:43.198654	2405.0	-86.5

SWEEP DATA FROM RFE → Logged as →

The timestamp of each logged row was as crucial as the spectral data contained in the row because measurement data would be joined with spatial data from the drone using timestamps. Initially, the apparatus included a GPS module connected to the Raspberry Pi with a serial GPIO connection; this enabled us to log spatial and spectral data collected simultaneously in the same row because all the data was flowing through a single script on the Pi. However, positional data

18

collected using the Pi was extremely unreliable compared to the robust data logged on the P3A; using the drone as source of data then necessitated having to join data collected on separate systems.

One implementation challenge that presented itself was the lack of a real-time clock (RTC) on the Raspberry Pi, which necessitated manual clock alignment before measurement could begin. Due to the omission of an RTC, the Raspberry Pi defaulted to its factory time setting at every boot and kept time from that point using a quartz crystal. On the other hand, the P3A logged accurate time in the UTC-3 timezone. To align the Raspberry Pi, the Network Time Protocol, or `ntp`, synchronization service had to be invoked when the Raspberry Pi was connected to a network over Ethernet. Once synced and disconnected from the network, the Pi had to remain on so it would not reset to its factory default time.

2.7. Designing and Mounting the Apparatus

The receiver apparatus had to be mounted to the P3A in order to take measurements at various heights in a 3-D space. Unfortuantely, the P3A is not designed with payload capacity as primary consideration; it is rated as having a low carrying weight of 300g, in addition to its internal components. The more weight the P3A is carrying, the shorter it's flight time and reliability is. Since the P3A, is sold as a photography device, it comes with a camera gimbal assembly already mounted under the drone. The first step was to detach the camera gimbal from the underside of the drone, revealing the mounting bracket attached to the body of the P3A. With the gimbal removed, the P3A could then be operated in the same way as before; the only difference in functionality was that no video was streamed from the aircraft to the controller. Manual and automated flight modes continued to funciton as described previously.

Once the gimbal was removed, several design choices regarding the design of the apparatus presented themselves. The measurement apparatus consisted of the three components: the Raspberry Pi, the RF Explorer spectrum analyzer and the Anker lipstick battery needed to power the Pi. One option was to secure these components to the body of the P3A separately; however, the Pi needed to be connected via USB cables to the other devices. Rather than have the devices taped individually

around the aircraft, requiring loose wiring to connect them, we chose to create a lightweight mounting platform that would securely fix to the now vacant bracket formerly occupied by the camera gimbal.

[MOUNTING IMAGE]

A lightweight, balsa wood board was cut to function as the platform for the mounting apparatus. Then, space constraints imposed by the P3A had to be accounted for while considering how to lay out the components of the apparatus. First, in order to attach the board to the bracket, 2 inch bolts with 1 inch spacers were inserted into the openings on the bracket; holes were cut out of the balsa board and the board was fixed in place with washers. This construction was necessitated by the placement of the bracket holes. Laid next to each other, the Pi and the RFE were simply wider than the width of the bracket holes, where bolts would have to secure the board to the drone. Instead, one of the devices would have to go on top of the board and the other on the bottom, which is why the spacers were introduced. Another constraint that presented itself in this revised design was the P3A's VPS sensor attachment, which was fixed to the bottom of the drone towards the rear of the craft and protruded out from the body. Further, if the VPS sensors were covered in any part by the measurement apparatus, optical stabilization would be disabled and the drone might not produce as precise spatial data.

The RFE had its antenna fixture at its headDd and its mini USB output at its tail Because the VPS protruded from the body of the drone and required unobstructed line-of-sight with the ground, the RFE then had to be fixed to the bottom of the board with its antenna oriented towards the front of the craft. The Pi had to be placed on top of the board with its USB ports at the front of the craft, again due to the protrusion of the VPS. A short mini USB cable with right-angle connectors (to minimize sensor obstruction) connected the two components. Additional holes were cut in the board to allow a zip tie to securely tie the components to the board. Lastly, the Anker lipstick battery couldn't fit on the board due to the legs of the craft; instead, it was fixed vertically on the leg of the craft closest to the Pi's power input port.

2.8. Initial Testing and Revised Design

The total weight of the measurement apparatus was [380 g] and the mounting fixture weighed in at [10 g]. This exceeded the load carrying capacity of the P3A significantly. However, we proceeded to test the flight performance of the mounted P3A to determine the extent to which flight would affected. The drone performed surprisingly well with this excessive load, with two exceptions. First, since the apparatus was mounted on the underside of the drone, a noticeable "pendulum" effect occurred where the drone swung slightly past it's stopping point at waypoints. This was more pronounced when flying at heights > 20 ft. As successive tests were performed, the drone began to struggle maintaining its height, likely due to insufficient power from a depleted battery.

Figure 9: Flight paths of two test flights with a maximum altitude of 21 ft, one where the measurement apparatus was mounted without modifying the components (yellow) and one after the apparatus was made lighter by 140 g (red). Note the increased stability of the flight path and the more precise turns.
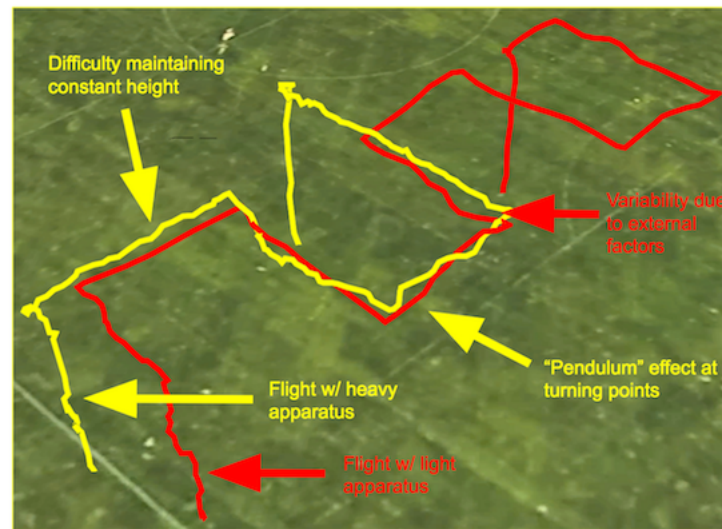
A few measures were taken to reduce the weight of the measurement apparatus. The RFE ships with a metal case enclosing its internal circuitry. Additionally, the RFE model used in this project supports measurements in the 900 MHz band as well as the 2.4 GHz band and included specialized antennas for both bands. Both the enclosing case (X g) and the the 900 MHz band antenna (X g) were removed to minimize weight. The final iteration of the measurement apparatus weighed X grams, well under the 300 g load carrying capacity and 60 g above the weight of the included

camera gimbal.Another adjustment made was increasing the speed of the drone in the Litchi app from 1 mph to 2 mph; this configuration resulted in the drone maintaining a constant height more easily.

3. Putting it Together: Building the Point Cloud (Proof of Concept)

Now that the receiver apparatus was mounted on the drone, it was time to see if the system as a whole could generate the desired point cloud of signal propagation. Spatial data would be collected from the drone's flight logs and joined with spectral data gathered from the measurement apparatus. Both systems would have to work in tandem to produce the data we were seeking.

It was imperative to perform a proof of concept (PoC) test run to validate the functionality of the system and our experimental methodology, and also discover the limitations of the system. To accomplish this, we decided to perform an experiment on an open field without the transmitter component. The transmitter was omitted because we wanted to focus on testing the built components of the project in isolation so it was easier to diagnose issues and observe the limitations of our spectral measurement apparatus.

3.1. Setup and Flight Planning

A set of flight paths using the "lawn mowing" approach across various heights yielded maximum coverage for consistent data collection and aggregation in 3-D. The mowing approach allows us to systematically sample points in a 2-D plane, as the drone flies in long, parallel strips spaced a few feet apart from one another. Popular drone geographic mapping services such as Skycatch Commander also use this technique to generate a consistent sampling of space. Since our goal is to construct a point cloud in 3-D, we can perform repeated mows at various heights to construct a series of spatial slices which can be stitched together to yield a sampling of 3-D space. Te first step to building the PoC point cloud is to create a 2-D flight path, or slice, as a Waypoint mission in the Litchi app. We can then modify the height of the result waypoints to gather slices at various elevations.

Due to data on the drone and apparatus being joined on timestamps, the Rapberry Pi's clock had

Figure 10: The "lawn mowing" approach, as viewed from above and from the side. A short lawn mowing flight mission created in Litchi for the proof-of-concept test run.
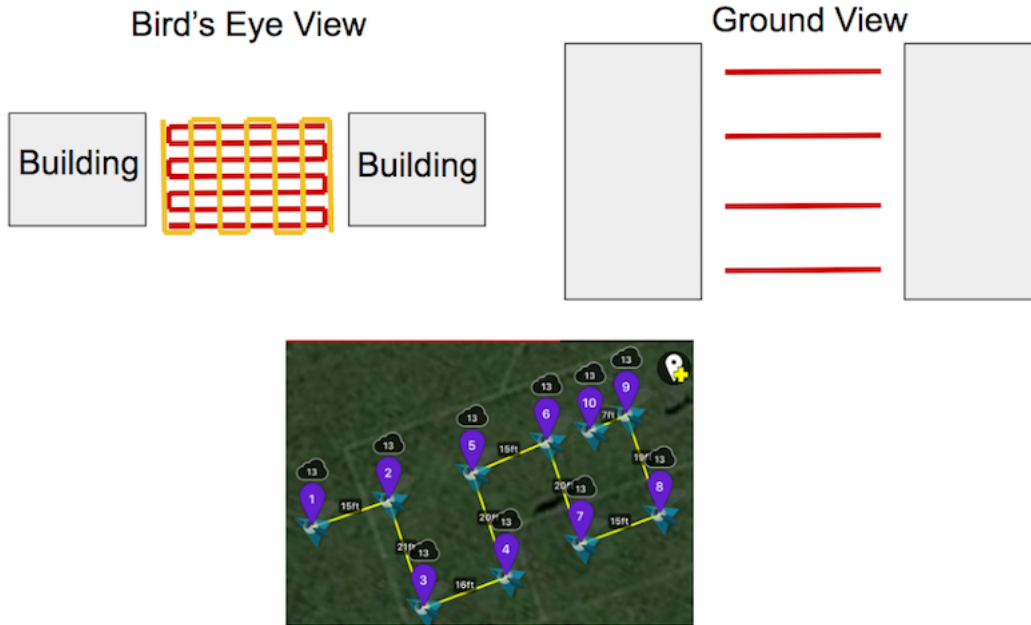
to be aligned to UTC-3 time (see Measurement and Logging Script) and the apparatus had to remain booted from alignment to the end of testing. In order for the Pi to log aligned data throughout the PoC, it had to be synced over a network connection and then left on en route to the testing location, where it would not be possible to connect the Pi to a network. Once the Pi was aligned, it was disconnected from the network and the measurement script was ran. This resulted in lots of extraneous data being logged when the apparatus was in transit; since the Pi had 32 GB of onboard SD card storage, however, storing large amounts of data was not a major issue.

Upon reaching the testing site, the drone was flown in a lawn mowing pattern at three heights with the RFE measuring in the 2.4 GHz ISM band. The minimum frequency of the RFE was 2406 MHz and the maximum was 2486 MHz. A 2dBi antenna tuned for 2450 MHz was connected to the RFE. Flights would be performed at 6 ft, 11 ft, 17 ft and 23 ft with the drone flying at 3.5 mph. Five flights would be performed autonomously at each height to get a thorough data set over which noise could be smoothed out. Autonomous flights were performed with the drone controller on, in case manual control of the aircraft had to be regained at any point.

In total, 18 flights across the 4 heights were performed. Despite the P3A having state-of-the-art

battery life, the drone reached critical battery levels and performed an auto-landing at the configured battery life threshold (see Limitations). The measurement apparatus was left on for the duration of testing, which lasted 2 hours and 1 minute, accounting for transit, setup and flight time.

3.2. Processing and Joining Data: Specifics

Spatial data collected from the drone flights and spectral data logged by the measurement apparatus had to be extracted, aligned and joined with each other to output the point cloud. Flight logs, each corresponding to a single flight, were downloaded from the Airdata UAV repository while spectral data was collected in a single large file on the Raspberry Pi. The Pandas data analysis framework was used to read the data sets from files into an iPython notebook; visualization was done with a combination of matplotlib and plotly.

Since data sets have to be joined on equal column values, the timestamps in spatial and spectral had to be aligned to a common format; for this project, timestamp columns were aligned to the nearest 100 ms and joined using an inner join. The drone flight logs saved rows of data with a 10 Hz update rate. Time information was split across two columns, with a "flight_time" column specifying how long (in milliseconds) the flight corresponding to the log file had been in progress and a "datetime(utc)" column logging the UTC time with second precision. Since flights were performed autonomously, rows in the flight log corresponding to auto take-off and landing flight states were filtered out. The resulting data set captured the entirety of the drone's waypoint mission flight. Flight time values were rounded to the nearest 100 ms and concatenated with values from the datetime column to form a consolidated timestamp field that could be joined on. Similarly, the RFE had an update rate of 5 Hz and each row in the measurement log file was timestamped with microsecond precision. These timestamps were also rounded to the nearest 100 ms. Filtering was performed on the logged spectral data so that only the rows that were logged during flight testing were processed in Pandas.

Once timestamps were aligned across the datasets, an inner join was performed to join the data sets. An inner join was preferred so that empty rows resulting from an outer join with only spatial

24

Figure 11: The complete data processing pipeline required for this project. Left: Spatial data processing on flight logs gathered from Airdata UAV and the Litchi app. Data from each flight is passed through this pipeline and appended to generate one table with all flight data. Right: Spectral data pipeline from the measurement apparatus. Inner join results in one table combining spatial and spectral data.
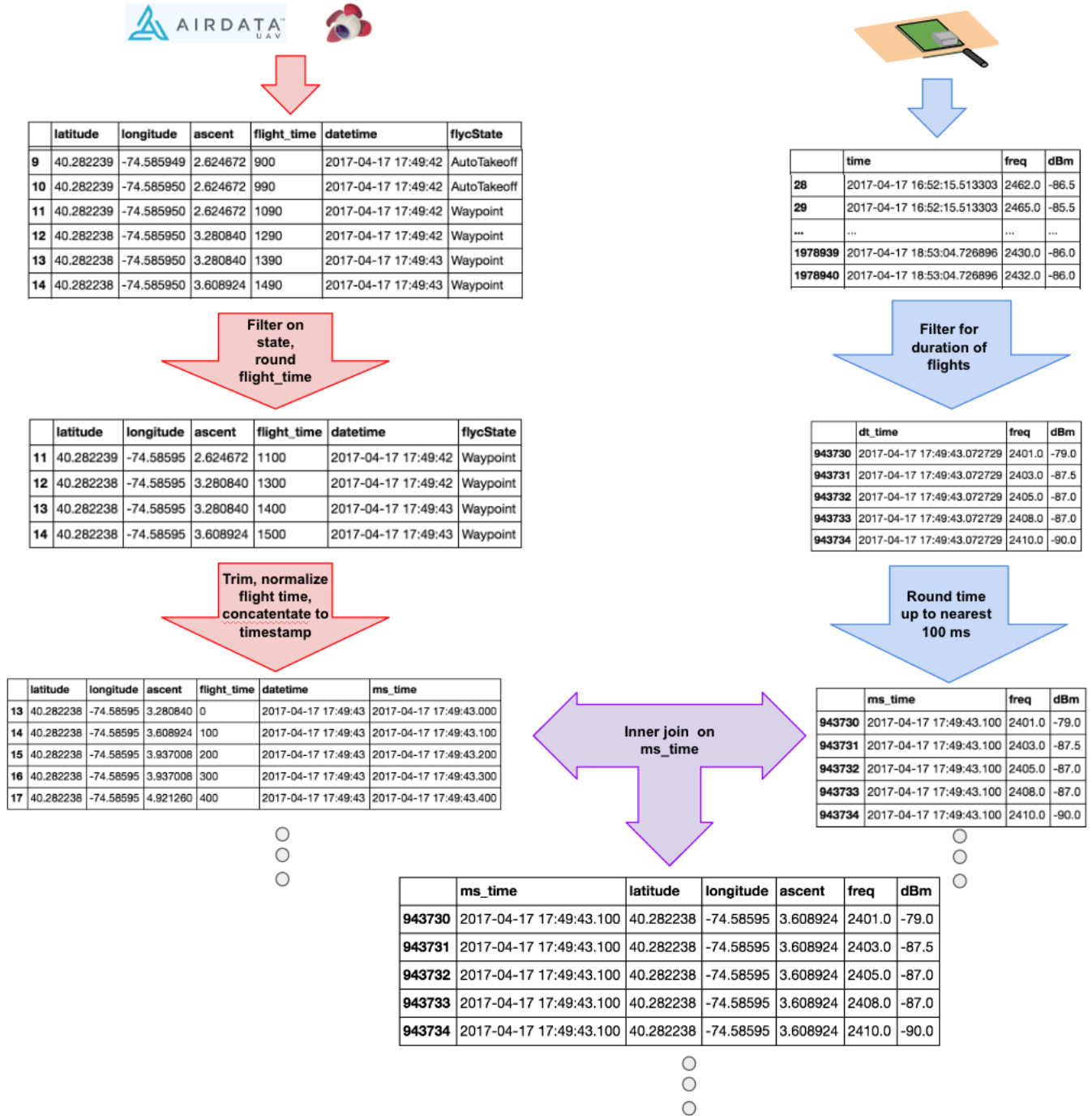
or spectral data could be avoided in practice. Aligning the timestamps with each other resulted in

> 99% of spectral data rows being included in the final joined data set. The size and update rate

of the resulting dataset was limited by the slower update rate (5 Hz) of the RFE. Since the P3A was flown at a speed of 3.5 mph, an update rate of 5 Hz translates to consecutive data points being spaced about 1 ft apart.

3.3. Visualizing the Point Cloud

Once the spectral and spatial data were joined, a few small tweaks and additions were needed to add context to the dataset. The latitude and longitude columns of the dataset were normalized to feet for consistency with the height data with the minimum values corresponding to 0 ft. Further, before plotting, an additional row with populated spatial fields and null spectral fields was added to indicate the position of the drone controller (and operator) relative to the measured space.

With these modifications, the joined data was ready for visualization and analysis. A 3-D scatter plot[1] was constructed using Plotly, where points were colored relative to the maximum and minimum signal strength values measured during the test flights.

3.4. Analyzing the PoC Propagation Data

Data collected during the PoC was gathered in the absence of a dedicated transmitter, so the point cloud measurements were representative of two sources of noise: ambient noise from the environment and transmissions from the drone controller. While the PoC was successful in terms of validating the functionality of the implementation and data processing pipeline, it also revealed large sources of noise which necessitated hardware changes moving forward.

As mentioned earlier, the 2.4 GHz ISM band is overcrowded due to the many ubiquitous consumer electronic devices operating in it. Because of this, there was a consistent ambient noise level across the 2.4 GHz spectrum. The RFE is sensitive to signals as weak as -120 dBm, but signal readings of -80 dBm were extremely common across the 2.4 GHz band. For example, the mean signal strength detected at 2442 +/- 2 MHz was -87 dBm, with a σ of just 2.9 dBm. In the point

---

[1]Plots produced for this work are interactive and will be hosted online using plotly at links in the footnote, pointed to by figure captions.

Figure 12: Visualizing a point cloud of signal strength measurements from the 18 flights performed for the Proof-of-Concept test run. Note the flight pattern in the "Bird's Eye View", which is the real world resulting from the Waypoint mission in Fig 10. Flights were performed at heights of 8 ft, 13 ft, 20 ft, 26 ft.
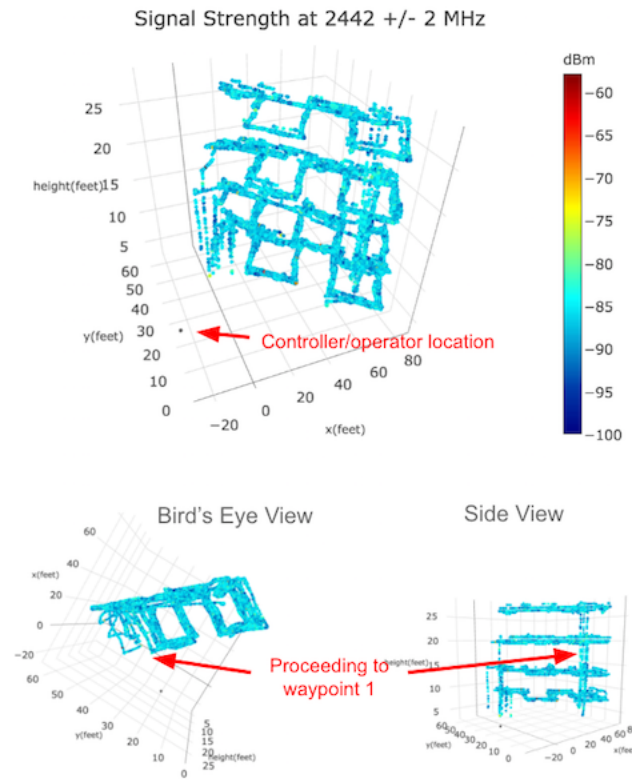
cloud, we can visualize this ambient noise as a steady sea of interference present across all spatial measurements.

Beyond a base noise environmental noise level, the P3A and its controller transmit their own wireless signals for control and data transmission purpose. Since the P3A is designed with photography in mind, it uses LightBridge, DJI's proprietary data transmission protocol, to beam a live video feed from the camera to the controller. LightBridge operates in the 2.4 GHz ISM band and the P3A begins transmitting data when it establishes a preliminary connection with the controller. The frequency channel over which Lightbridge is operating can be coarsely configured; for all the experiments, Lightbridge channel 20 (2471MHz - 2482MHz) was selected. The aircraft and the controller transmit over this channel continuously and this is evident in the constantly high (> -40 dBi) point cloud measurements for frequencies falling withing the range of Lightbridge channel 20. Despite the built-in camera being disconnected and no video being available, the P3A continues

27

to transmit data from the craft's telemetry and navigation systems in the configured Lightbridge channel. While the data transmission frequency range can be manually set, the P3A receives remote control signals from the controller at non-configurable, variable frequencies throughout the 2.4 GHz ISM band. A frequency hopping, spread spectrum (FHSS) method is used for control transmissions, where data is sent over rapidly changing frequencies between the craft and the controller, randomly hopping across the 2.4 GHz ISM band. While these transmissions are drowned out by constantly strong signals in the data broadcasting frequencies, they are briefly observable in the mid-to-lower reaches of the RFE's measured spectrum. For example, at 2412 +/- 2 MHz, the mean signal strength measurement was -87.6 dBm with a $\sigma =2.9$ MHz; however, there were 58 measurements out of 28695 that registered greater than -70 dBm.

Figure 13: Top: Point cloud representations of measurements at 2412, 2442, and 2472 MHz. Note that 2472 MHz is within the Lightbridge channel setting of the P3A controller. Evidence of high measurements due to the FHSS control signal are highlighted at lower frequencies. Bottom: Signal strength distributions at the various frequencies. Note the persistent ambient noise profiles across the 2.4 GHz band.