DOT/FAA/TC-22/41

Federal Aviation Administration
William J. Hughes Technical Center
Aviation Research Division
Atlantic City International Airport
New Jersey 08405

Rotorcraft Wire Strike Prevention and Protection: Wire Database Creation, Sensor Package Design, Wire Cutter Modeling, Data Fusion, and Helicopter Operators' Inputs

February 2023

Final Report

U.S. Department of Transportation
Federal Aviation Administration

NOTICE

Technical Report Documentation Page

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

1. Report No.
DOT/FAA/TC-22/41

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
Rotorcraft Wire Strike Prevention and Protection: Wire Database Creation, Sensor Package Design, Wire Cutter Modeling, Data Fusion, and Helicopter Operators' Inputs

5. Report Date
February 2023

6. Performing Organization Code
ANG-E272

7. Author(s)
Gabriel Achour (GT), Caleb Harris (GT), Matthew Dvorsky (ISU), Alexia P. Payan (GT), Brook Wheeler (FIT), David Eisenmann (ISU), Reza Zoughi (ISU), Mohammad Al Qaseer (ISU), and Dimitri Mavris (GT)

8. Performing Organization Report No.
DOT/FAA/TC-22/41

9. Performing Organization Name and Address

Aerospace Systems Design Laboratory
The Daniel Guggenheim School of Aerospace Engineering
Georgia Institute of Technology
620 Cherry Street, Atlanta, GA 30332

10. Work Unit No. (TRAIS)

11. Contract or Grant No.

12. Sponsoring Agency Name and Address

Federal Aviation Administration
William J. Hughes Technical Center
Atlantic City International Airport
Atlantic City, NJ 08405

13. Type of Report and Period Covered
Final Report

14. Sponsoring Agency Code

15. Supplementary Notes
The Federal Aviation Administration Aviation William J. Hughes Technical Center Research Division COR was Traci Stadtmueller

16. Abstract

Wire strikes have been one of the most common sources of helicopter accidents in the past seven years. The sizes of poles and wires make it difficult for rotorcraft pilots to see them in-flight. This motivates the need to provide pilots with various tools and information to prevent accidents related to wire strikes. In this report, we present different ways to enhance the safety of helicopters when operating at altitudes where wire strikes are possible. The first way enables the prediction of the wire grid from street view and satellite imagery using deep learning techniques. The outcome is a dataset of predicted wire locations that may be displayed in 2D or 3D on moving maps of electronic flight bags or in the cockpit avionics. The second way of preventing rotorcraft wire strikes is the development of a 360-degree radar system able to detect wires in-flight at a minimum distance of 50' from any part of the rotorcraft. The third way is to fuse the information from the predicted wire database with the measured data from the sensor in order to improve the predicted wire locations in real time. A fourth way is to develop a new wire cutter system able to cut various sizes of cables when installed on lightweight helicopters. To do so, we develop a physics-based modeling and simulation environment where various types of helicopters, wires, and cutter systems may be implemented and assessed. Finally, a questionnaire was created and distributed to helicopter operators to determine how to prevent wire strike incidents and accidents based on their operational experience.

17. Key Words

18. Distribution Statement

This document is available to the U.S. public through the National Technical Information Service (NTIS), Springfield, Virginia 22161. This document is also available from the Federal Aviation Administration William J. Hughes Technical Center at actlibrary.tc.faa.gov.

19. Security Classif. (of this report)
Unclassified

20. Security Classif. (of this page)
Unclassified

21. No. of Pages
172

22. Price
Unclassified

Contents

Figures

Tables

Acronyms

Acronym	Definition
API	Application programming interface
CNN	Convolutional Neural Network
DOF	Degree-of-freedom
EFB	Electronic Flight Bag
EM	Electromagnetic
FAA	Federal Aviation Administration
FCE	Fundamental cylindrical elements
FM-CW	Frequency-modulated continuous-wave
GCS	Google cloud storage
GEE	Google Earth Engine
GPS	Global positioning system
GPU	Graphics processing unit
GSV	Google Street View
HAI	Helicopter Association International
MCC	Matthews correlation coefficient
ML	Machine Learning
NASA	National Aeronautics and Space Administration
NTSB	National Transportation Safety Board
PACE	Partnership for an Advanced Computing Environment
PFD	Primary flight display
RANSAC	Random Sample Consensus
RCS	Radar cross-section
SAR	Synthetic aperture radar
SVM	Support Vector Machine
UAV	Unmanned Aerial Vehicle
VFR	Visual Flight Rules
VNA	Vector network analyzer
WSPS	Wire Strike Protection System

Executive summary

Wire strikes have been one of the most common sources of helicopter accidents in the past seven years. The sizes of poles and wires makes it difficult for rotorcraft pilots to see them in-flight against the background. This motivates the need to provide pilots with more tools and information to prevent accidents related to wire strikes. In this report, different methods and technologies are presented to enhance the safety of helicopters when operating at altitudes where wire strikes are possible. First, we describe how to construct a wire grid from street view and satellite imagery using deep learning techniques. The wire grid provides the locations of utility poles and wires that may then be displayed on a map. We then show how to potentially display the predicted information in 2D and 3D on electronic flight bags. We also present the development of a radar sensor for lightweight helicopters that may be used to detect wires in-flight, and we describe the indoor and outdoor testing environment of such a radar system. Finally, we detail the methodology that could be used to fuse information obtained from the radar sensor with the wire location database to improve the prediction of the locations of poles and wires in real time. Additionally, we present the creation of a physics-based modeling and simulation environment to study the dynamics of wire strikes by modeling the helicopter system, the wire system, and the associated impact. A final aspect of the research presented in this report focuses on helicopter operators' input regarding wire strike incidents and how to prevent them, in order to confirm that the tools and techniques or methods showcased in this report answer the operators' needs in terms of safety.

1 Introduction

Wire strikes are a major source of helicopter accidents. A Federal Aviation Administration (FAA) study showed that wire strikes accounted for five percent of total helicopter accidents from 1963 to 2008 (Stadtmueller, 2018). Among the five percent, one-third of the wire-strike accidents involving civil helicopters between 1994 and 2004 were fatal and mostly occurred during clear, daytime, Visual Flight Rules (VFR) conditions. Therefore, the FAA has made wire strike accident prevention a major priority for rotorcraft safety.

Rotorcraft operations include a wide range of missions such as aerial seeding, medical response, and power line monitoring. These missions often require flight at low altitude, which increases the risk from obstructions in the flight path. Consequently, wire strikes are a major source of accidents, with 214 wire strike accidents and 124 fatalities between 2005 and 2018 according to an FAA study (Stadtmueller, 2018). These accidents occur not only in cases of visual impairments like fog or sun glare, but when there is a lack of flight planning for the mission, or when the pilot is simply unable to see the wires. Therefore, there is a critical need to protect pilots from these collisions.

Wire strike accidents often occur when the pilot is unable to locate the wire before the collision occurs. This is related to both the knowledge of where the wire is before-flight and the ability to see the wire in-flight. Currently available aeronautical charts do not indicate complete networks of power lines nor all the other similar aerial obstructions. The precise location of wires and cables should always be indicated to pilots as wires may not be visible in-flight. Even if the locations of wires/cables are known beforehand, without the precise location always indicated to the pilot, wires can easily become invisible due to lighting conditions or the background landscape. The knowledge of where the wires are is particularly critical around high-risk areas such as mountainous regions and bodies of water. Also, it is revealing that 60% of the wire strike accidents were categorized under Part 91 General Aviation flight operations (Stadtmueller, 2018). General Aviation (GA) operations are not subject to the same restrictions as commercial operations and involve many recreational pilots. Therefore, it could be that the solution proposed have not had a major effect on wire strike events. In addition, GA pilots may be less likely to have access to advanced technologies and more detailed flight planning that would help the prevention of such accidents.

An extended investigation performed during a previous phase of this research and available in Chandrasekaran et al. (Chandrasekaran R. , Payan, Collins, & Mavris, 2020) further clarified the feasibility of developing a wire database to improve rotorcraft safety. Work was done to further specify the problem by identifying flight conditions that lead to wire strikes, and to define the challenges and limitations of a wire database. After these were found, a focused literature survey was completed in the various methodologies of forming the database.

Wire strikes can occur by many causes, including flight conditions, failed maneuvers, and unknown wire obstructions. The scenarios in which they occur are often very difficult to predict. A subset of wire strikes from the National Transportation Safety Board (NTSB) database[1] were used to clarify these types of scenarios, each of which resulted in one or more fatality. These included accident numbers: ERA20FA012 (Salem), WPR13GA128 (Eureka), and CEN13FA295 (Cross Timbers). These accidents give good insight as each has very different circumstances which caused it, and each would need varying changes to prevent. Further details are found in Table 1, Figure 1 and Figure 2.

Table 1. NTSB Accidents

	Salem	**Eureka**	**Cross Timbers**
Location	North Carolina; Farming Land	Nevada; DOI land	Montana; Forest
Time / Weather	Day; Clear	Day; Snowfall	Day; Clear
Aircraft / Pilot	Bell 206B; Single pilot	Bell 206B; Single Pilot	R44; Two Pilots
Conclusion	Pilot was aware of the power lines, but must have forgot or lost track of location. Distribution lines were not depicted on any available charts.	Pilot was aware of the lines, and those transmission lines were depicted on available charts.	Unknown circumstances for crash. Lines were not depicted on available charts.

---

[1] https://www.ntsb.gov/_layouts/ntsb.aviation/index.aspx

Figure 1. Images of accident locations

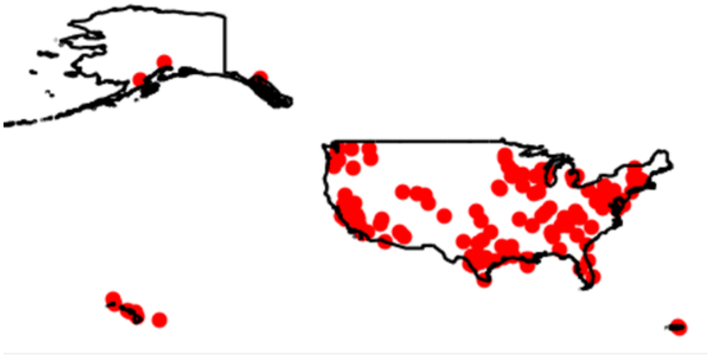


Figure 2. Diverse location of wire strike accidents across the United States

2 Feasibility of developing a wire database

Rotorcraft collisions with wires, particularly power lines, are difficult to predict and often result in fatalities. For this reason, there is a push to provide pilots with additional information regarding wires in the surrounding environment of the helicopter. However, the precise locations of power lines and other aerial wires are not available in any centralized database. This research considered the development of a wire database in two stages. First, poles and towers were detected from aerial imagery using deep learning and two methods of object detection (tile classification and semantic segmentation), were compared. Second, the complete grid network was predicted using a weighted graph search. The resulting two-step framework resulted in a complete medium and high-voltage grid representation. Experiments were conducted in Washington D.C. using openly available datasets. Preliminary results were promising and showed that utility pole locations may be predicted from satellite imagery using deep learning methods. The goal was to apply the proposed framework to larger regions of the U.S. to map various wires/cables that may endanger rotorcraft safety.

While there is important work in wire strike protection systems and sensors to detect wires onboard, as shown in the detailed review by Chandrasekaran, et al. (Chandrasekaran R. , Payan, Collins, & Mavris, 2020), the primary goal is to provide pilots with accurate, detailed, and adaptable information. This work proposed the creation of a centralized wire database to address this problem, which pilots can use before or during flight operations to avoid wires and similar obstacles.

Databases for power lines and other aerial towers do exist, though in limited scale and detail. For instance, a portion of the United States electrical grid can be found from the U.S. Department of Homeland Security open database (U.S. Department of Homeland Security, 2020). This includes data on electric substations and electric transmission lines. However, it is limited to substations and transmission lines for voltages of above 69 kilovolts. Furthermore, it is noted that coverage is not complete even under these limitations. Three example cases of wire strike accidents reported by the NTSB are shown in Table 1. Two of the cases did not have any documented wires and although one case did, this was not informative enough to prevent the accident.

Additional data in the form of open-source database, such as the OpenStreetMap[2] database can be a powerful supplementary source for various types of geospatial information. However, this information must be used cautiously as the verification process of the publicly sourced data entered in this database is below the required standards for use in aviation.

Nevertheless, any centralized wire database must be highly accurate to provide meaningful information and must be capable of updating in accordance with construction and grid changes. Also, the database must be scalable since accidents occur across the entire United States.

In recent years, the capability of remote sensing and object detection from aerial and satellite imagery has improved. This is in part because of the advancements in deep learning and in a wider availability of aerial and satellite imagery (Zhu, et al., 2017). Example applications can be seen from competitions such as the Defense Science and Technology (DSTL) Satellite Imagery Feature Detection and the SpaceNet Challenge. Furthermore, this field of research has already made progress into electrical grid detection and prediction, particularly from work by the Development Seed and World Bank (The World Bank, 2018), and by Facebook Engineering (Gershenson, Rohrer, & Lerner, 2019). However, these two studies focused on the mapping of high-voltage lines using a single dataset of high-resolution satellite imagery. Therefore, new techniques are needed to expand the potential to medium-range wires and to utilize multiple types of satellite and aerial imagery. Furthermore, this information needs to be collected and processed in a way which best provides pilots and crew with information before and during flight.

---

[2] https://www.openstreetmap.org/

It can be difficult to identify poles from satellite imagery if the area of interest is crowded with trees or if the imagery resolution is low (Cetin, 2009; Matikainen, et al., 2016). This phenomenon is illustrated in Figure 3. As such, recent studies focus on the detection of utility poles and other objects from street view imagery (Tang, Wang, Majumdar, & Rajagopal, 2019; Krylov, Kenny, & Dahyot, 2018; Zhang, et al., 2018). The poles and wires are more easily detected as the imagery was taken much closer to the objects of interest. However, street view imagery is limited to available street locations and consequentially off-road wires cannot be predicted.

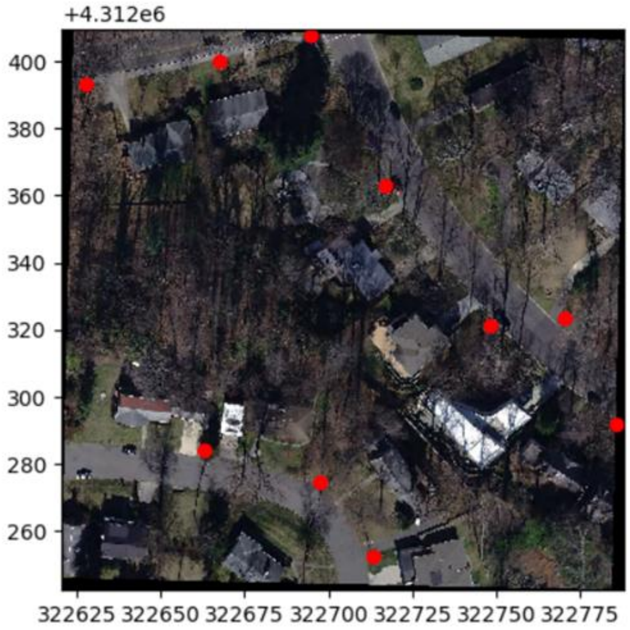


Figure 3. Utility poles (red dots) difficult to distinguish when mixed with trees

A two-step process was defined to address these difficulties and limitations. The first step involves the detection of poles and towers using satellite and street view imagery. Two methods were compared for object detection on satellite imagery: tile classification using the Xception network and semantic segmentation using the UNET architecture network. For the street view imagery, the convolutional neural network Inception Resnet V2 used as a classifier was leveraged. The UNET and Inception Resnet V2 network were then combined to provide a list of predicted locations of wires and poles. The second step used the most trusted detections to predict the location of the local grid network. This was done by using known information, such as a substation or power plant, and linking detections together through a weighted graph search. Together, the two steps provided a pipeline to generate a complete electrical grid database.

2.1 Background

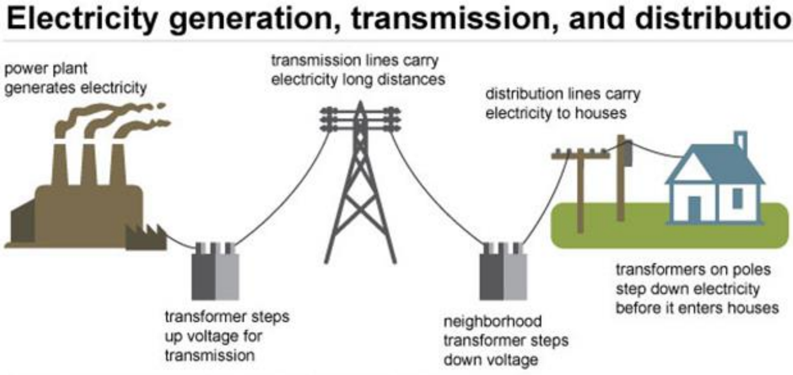
2.1.1 The electrical grid

The electrical grid in the United States is divided regionally with a distributed hierarchy of authority and management and features a combination of high voltage transmission lines, medium voltage distribution lines, and low voltage utility lines (Hoffman, Streit, Gilstrap, Amin, & DeCorla-Souza, 2015). An example of the electrical grid structure is shown in Figure 4. As of 2016, the grid included over 6 million miles of transmission lines and almost 500,000 distribution lines (Warmick & Hoffman, 2016). In addition, the power lines and poles of the resulting grid vary in shape and size. Therefore, there is no simple technique to detect or predict the grid. In the past, power lines have been detected and mapped on small scales for the purpose of line inspections and vegetation management, but not with the goal of knowing and keeping up with the precise location of these structures.

A primary distinction between the type of power lines is the operating voltage level. The operating voltage is divided into four categories: low, medium, high, and extra high, as described in Table 2 (U.S. Department of Energy, Office of Electricity Delivery and Energy Reliability, 2015). The grid is structured as an integrated system of power plants, transformers, transmission lines, distribution lines, and users.

Table 2. Power line descriptions for operating voltage level

Operating Voltage	**Terminology**	**Description**
Low	LV; utility	Less than 1000 volts; used for connection between a residential or small commercial customer and the utility.
Medium	MV; distribution	Between 1000 volts (1 kV) and 69 kV; used for distribution in urban and rural areas.
High	HV; sub-transmission less than 100 kV; sub-transmission or transmission at voltages such as 115 kV and 138 kV	Used for sub-transmission and transmission of bulk quantities of electric power and connection to very large consumers.
Extra high	EHV	From 345 kV, up to about 800 kV; used for long distance, very high-power transmission.



Source: Adapted from National Energy Education Development Project (public domain)

Figure 4. Structure of U.S. electrical grid

A database from the Homeland Infrastructure Foundation–Level Data (HIFLD) (U.S. Department of Homeland Security, 2020) which includes power plants, substations, and electrical transmission lines describes the electrical grid as:

> "Electric power substations primarily associated with electric power transmission as represented by facilities and equipment that switch, transform, or regulate electric power at voltages equal to, or greater than, 69 kilovolts. Substations with a maximum operating voltage less than 69 kilovolts may be included, depending on the availability of authoritative sources, but coverage of these features should not be considered complete. Geographic coverage includes the United States and the U.S. Territories."

2.1.2 Power line detection

Power line detection through remote sensing has been tested in the past, largely for power line inspection tasks, and a majority of these power line detection methods involve onboard sensing (Zhang, Yuan, Li, & Chen, 2017). However, a small set of literature has sought to document the power lines or similar objects from satellite and aerial imagery as described in the survey by Matikainen et al. (2016). Two documented examples used computer vision techniques, such as epipolar constraints (Zhang, Yuan, Li, & Chen, 2017) and Radon transform (Yan, Li, Zhou, Zhang, & Li, 2007) to specifically detect the power lines. These methods showed some level of success using fundamental computer vision techniques; however, they are time-consuming to implement, computationally expensive to run, and not successful under high levels of noise or obstruction. Additional research has been applied to other types of detection and sensing methods that show promising results.

In computer vision, feature detection is a key task for many applications. A wide variety of detection algorithms such as Binary Robust Invariant Scalable Keypoints (BRISK), Features from Accelerated Segment Test (FAST), and Speeded Up Robust Feature (SURF) have been implemented (Sahin & Kemal, 2014). These algorithms try to identify and recognize patterns or shapes to locate the feature of interest. However, for this effort, the poles and wires as seen from aerial views would be too small and difficult to identify using the mentioned algorithms, especially in a dense urban area where other features such as trees and building antennas would have similar shape patterns.

Deep learning architectures, specifically Convolutional Neural Networks (CNN), have shown to be a very promising approach for image classification (Krizhevsky, 2012). This approach is very flexible and scalable, hence adaptable to any image resolution and size. Deep learning in remote sensing has grown largely because of the improvement of Deep Neural Network (DNN) architectures, compilation of available datasets for training and testing, and the availability of processing power from graphics processing units (GPUs) (Tsagkatakis, 2019).

2.1.3 Grid network prediction

To address the problem of grid network prediction, first, it was beneficial to consider the information that was already known. For instance, the locations of at least a subset of the substations were known. In addition, the locations which are guaranteed to have power lines, such as around city centers, were also known. Furthermore, the proposed detection methods should provide additional locations. Next, if the physical environment of the grid network is defined as a graph, grid network prediction could be generalized as a graph search problem where path weights are allocated, with the goal to minimize the total weight of the graph. While there was no source of information to know the exact weight each graph edge should have, various features could be used to approximate or estimate the error from the true grid network path.

The many-to-many graph search was a problem addressed in a few areas such as map matching for predicting user movement on road networks (Storandt, 2019), the network competition problem for finding missing node edges in graphs such as social networks (Kim, 2011), and the many-to-many shortest path problem for optimized routes in road networks (Wagner, 2007). While each of these methods showed promising techniques for computational efficiency and speed for a graph network, the exact underlying structure of the grid network was unknown. Therefore, the work by Facebook Engineering (Gunning, Gershenson, Rohrer, & Lerner, 2019) approached the problem using a 2D grid in the area of interest and finding optimal paths using the path search method, Dijkstra's algorithm.

As previously discussed, there were two key examples in literature for grid network prediction by Facebook research (Gunning, Gershenson, Rohrer, & Lerner, 2019) and Development Seed (The World Bank, 2018). Development Seed utilized the Xception network for detection. However, instead of automating the network creation, a group of workers hand-labeled the network by drawing lines between pole detections. This was an expensive process in both cost and time. It is unlikely to have a full grid of the United States mapped and kept up to date with the human labelling processes. In another work, Zain (Zain, 2018) used a sliding window approach to detect the tower features, and similarly used direct lines to connect into a local transmission grid. These lines are automatically created but only within an area of interest with a few tower detections. Furthermore, this method was primarily interested in areas with many straight paths of power lines where vegetation encroachment occurs.

2.2 Problem setup

A clear outline for the methodology was formulated from the concern for rotorcraft safety around wires and the background provided from the extensive literature review. Previous research provides a starting point for utilizing various methods and techniques. For instance, the project from Facebook Engineering (Gershenson, Rohrer, & Lerner, 2019) concluded their work as follows:

> “We initially tried a computer vision approach to detecting MV infrastructure, using high-resolution satellite imagery collected during the day. But there were significant challenges with this method. For one, the shape and orientation of poles and structures are so diverse in form that even large image training sets were insufficient to generate high-quality labeled examples. In addition, vegetation, shadows, and nearby similar infrastructure made it hard to correctly identify MV grid lines. We decided to instead try a predictive modeling approach, using indicators of electrification.”

This was verified by using other classic machine learning (ML) techniques to predict locations of power poles. The Washington DC grid was separated into hexagons with an area of about 1000 square feet (about a 30 ft *radius*). The set of features included Boolean variables of whether buildings, roads, water, etc. were in the hexagon area. Additionally, other features such as terrain and relative height were used. Three models were used to train: support vector machine (SVM), decision tree classifier, and boosted decision tree classifier. The results shown in Figure 5 indicate that they were largely biased by the location of roads and therefore, there were clear false negatives. The unsatisfactory results led to the need for more advanced methods.

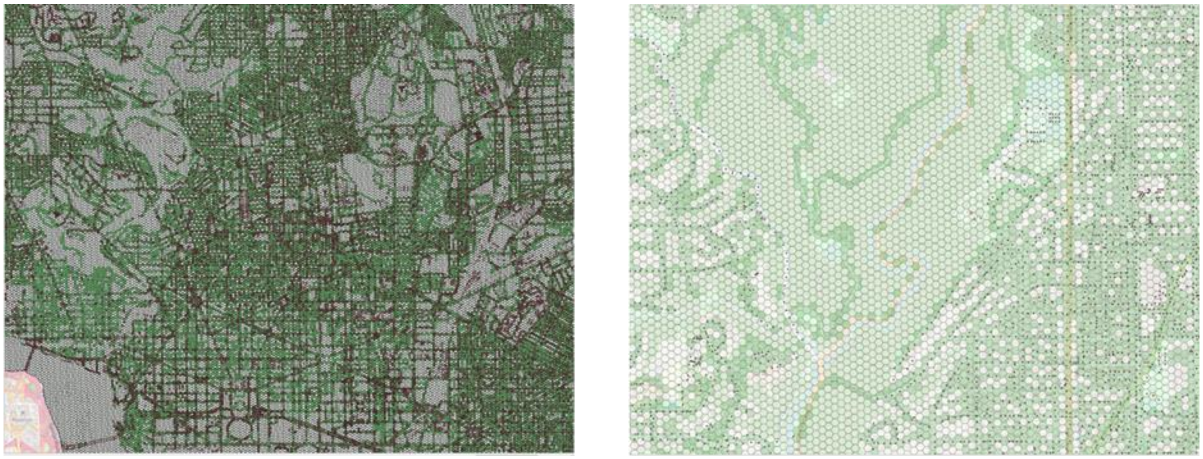


Figure 5. Visual results from SVM method

2.2.1 Deep learning convolutional neural networks

Based on the previous discussion and problem exploration, deep learning techniques with convolutional neural networks were chosen to predict the location of wire poles from satellite imagery and street view imagery. Satellite imagery datasets can be substantial in size, especially if high resolution imagery is leveraged. It was then clear that any data processing and ML training on this type of data can be computationally expensive, hence a data pipeline and environment needed to be chosen with care.

2.2.2 High-level data pipeline

The overall goal was to obtain a power line network from satellite imagery and street view imagery. To do so, two deep learning models using both sets of data are combined to predict the location of utility poles and then, a grid search algorithm predicts the location of the connecting wires between each predicted pole. Figure 6 depicts the high-level data pipeline used in this research. This project was performed in two phases. The first phase introduced the overall wire prediction pipeline consisting of two sub-steps: the pole location identification and the wire network prediction. The second phase augmented the pipeline by introducing another mean to detect the pole location enhancing the first sub-step of the first phase.

Identically, to any ML pipeline, the first step was the dataset preparation performed before the training of the model. This step was important to translate the raw data into normalized data that can be understood by the prediction algorithm. Moreover, in most cases, the target data was included in a separate dataset which was then merged and uniformized into a single dataset that could be used to train the model.

The second step was the actual implementation of the deep learning model. A training environment needs to be prepared to ensure the correct data flow during the training, and more importantly, a network architecture and its hyperparameters needs to be chosen. The model hyperparameters included the number of training epochs and the training batch size. Most of the time, their value needed to be tuned manually but a hyperparameter algorithm could also be used to perform the tuning automatically. After the training, the model could then be tested on new data and be applied to the areas of interest.

An additional step was necessary to extract the longitude and latitude of the predicted poles and wires depending on the architecture used, as each architecture can output a different prediction format. The outcome of this process was the ability to predict the presence of utility poles from satellite imagery and wires from street view imagery.

Once the pole locations were estimated, the next step was the use of a grid prediction model using inference to predict the wire map connecting the predicted poles.

The final step of this process was creating the visualization of the results for their implementation in an on-board display, an electronic flight bag (EFB) and then communicating the information to the pilot. The high-level data pipeline for wire and pole prediction is shown in Figure 6.

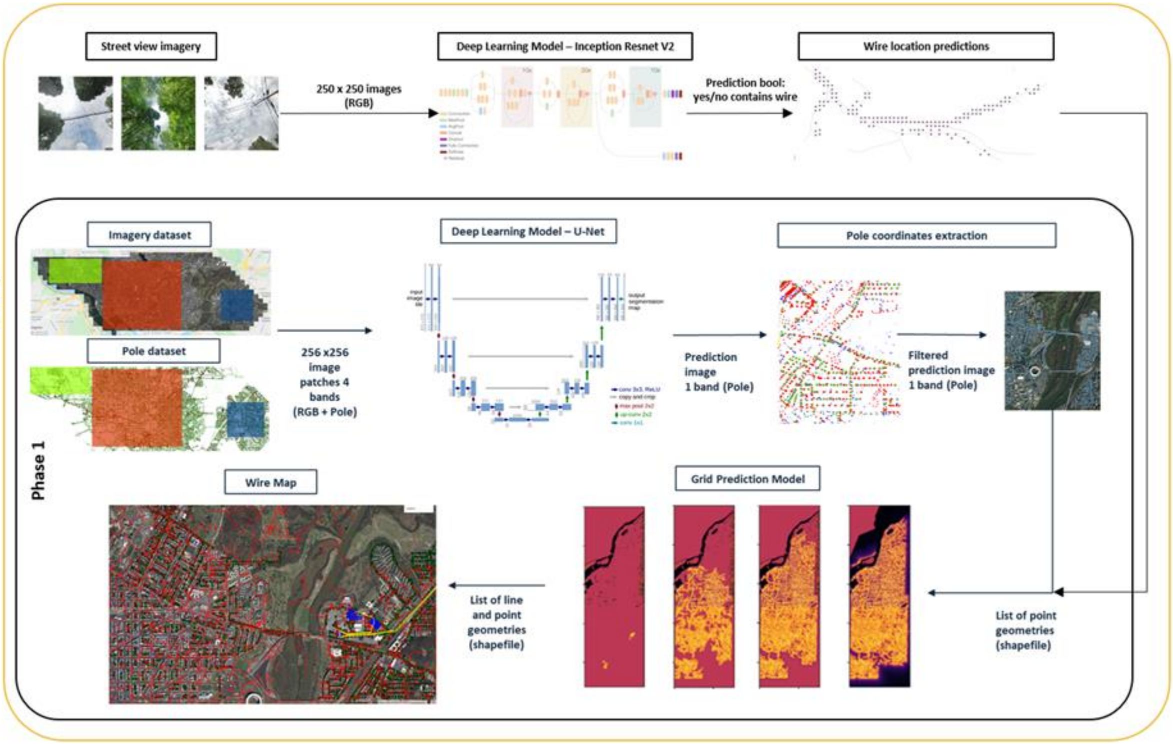


Figure 6. Overall pipeline

2.2.3 Datasets

In this work, the training dataset had three main components. The first one corresponded to satellite imagery as features to predict from, the second to street view imagery, and the third to pole locations as targets. While it can be relatively straightforward to obtain satellite imagery using tools like Google Maps or Bing, pole location data is not currently available everywhere in the USA. Moreover, high resolution satellite imagery is necessary but not acquirable from Google Maps or Bing. Fortunately, some states including Washington, D.C., Vermont, and New York publicly provide high resolution imagery along with tables of pole coordinate locations. The street view imagery can be acquired anywhere worldwide using the Google Street View application programming interface (API).

2.2.3.1 Satellite imagery

To enable the possibility for a deep learning model to learn from the dataset and recognize the pixels pertaining to a pole, the resolution for the imagery data needs to be sufficiently high. For example, the publicly available Landsat-8 dataset (30m resolution) would not be suited for this project, as shown in Figure 7.

In recent years, many companies and nations have increased the resolution and frequency of satellite imagery. Some commercial imaging satellites provide high-resolution imagery, such as

Worldview by Digital Globe[3] (Table 3). However, the cost of this data made it unfeasible for use in this research. For example, if the entire region of Washington, D.C. was required, then the total cost for the 179 km^2 would be approximately $2,506. At the estimated rate of $14 per km^2, the entire United States would be over $100 million.

Table 3: Commercial imaging satellites pricing

Comparing High- and Medium-resolution Commercial Imaging Satellites									
Satellite	**Launch Date**	**Swath Width (kilometers)[1]**	**Native GSD (meters)[2]**	**Output Resolution (meters)**	**Max View Scale[3]**	**Native Accuracy (meters)[4]**	**Bands**	**Bit Depth**	**Stereo**
IKONOS	Sept. 24, 1999	11.3	0.82x3.20	1x4[5]	1:2,500	15	pan + 4MS	11	yes
QuickBird	Oct. 18, 2001	18[6]	0.65x2.62	0.6x2.4	1:1,500	23	pan + 4MS	11	no
SPOT-5	May 3, 2002	60	5x10x20	2.5x5x10x20[7]	1:5,000	48	pan + 4MS	8	yes
WorldView-1	Sept. 18, 2007	17.7	0.5	0.5	1:1,250	5	pan only	11	yes
RapidEye	Aug. 29, 2008	77	6.5	5	1:12,500	23-45[8]	5 MS (no pan)	12	no
GeoEye-1	Sept. 6, 2008	15.2	0.41x1.65	0.5x2	1:1,250	5	pan + 4MS	11	yes
WorldView-2	Oct. 8, 2009	17.7	0.46x1.85	0.5x2	1:1,250	5	pan + 8MS	11	yes
Pleiades 1	Dec. 16, 2011	20	0.70x2.4	0.5x2	1:1,250	To be determined	pan + 4MS	12	yes

Primarily for this reason, the focus was on openly available datasets. The first and main focus of the work was the DC From Above[4] dataset of Washington, D.C. The data was provided in a single, compressed .jp2 file which could be organized into a 73x58 grid of images, each with dimensions 4096x4096. The organized images are shown in Figure 8. Each image, or tile, is a total of 16,777,216 pixels. The resolution is 0.08 meters per pixel, or 3.15 inches per pixel. Each image features three different bands corresponding to the RGB (red, blue, green) color code of each pixel. For example, if a pixel is yellow, then the data is encoded as [0, 256, 256]. In

---

[3] http://worldview3.digitalglobe.com/

[4] https://opendata.dc.gov/pages/dc-from-above

comparison, the other publicly available imagery from satellites and aerial flights are shown in Table 4.

Table 4: Comparison of openly available satellite imagery

Imagery	**Bands of Interest**	**Resolution**	**Accessibility**
DC From Above	RGBA	3-inch	Open-source (.jp2 compiled file)
Vermont Imagery	RGB	15cm	Open-source (.jp2 compiled file)
Atlanta Imagery[5]	RGB	10-inch	Open-Source (.jp2 batch of image files)
Copernicus Sentinel-2	10 Bands	10-meter	Earth Engine
NAIP	RGBA (Red, Green, Blue, Infrared)	1-meter	Earth Engine
LANDSAT-8	11 Bands	30-meter	Earth Engine

---

[5] https://gis.fultoncountyga.gov/apps/AerialDownloadMapViewer/

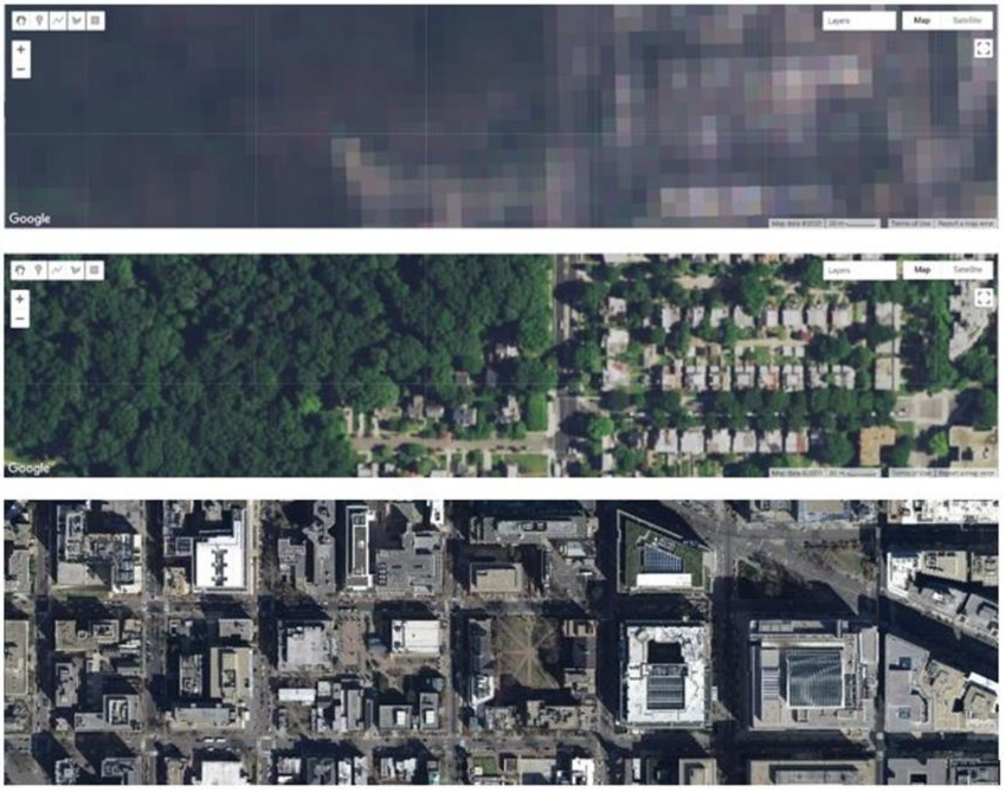


Figure 7. Comparison of aerial imagery datasets (top: Sentinel, middle: NAIP, bottom: Washington, D.C. from Above)

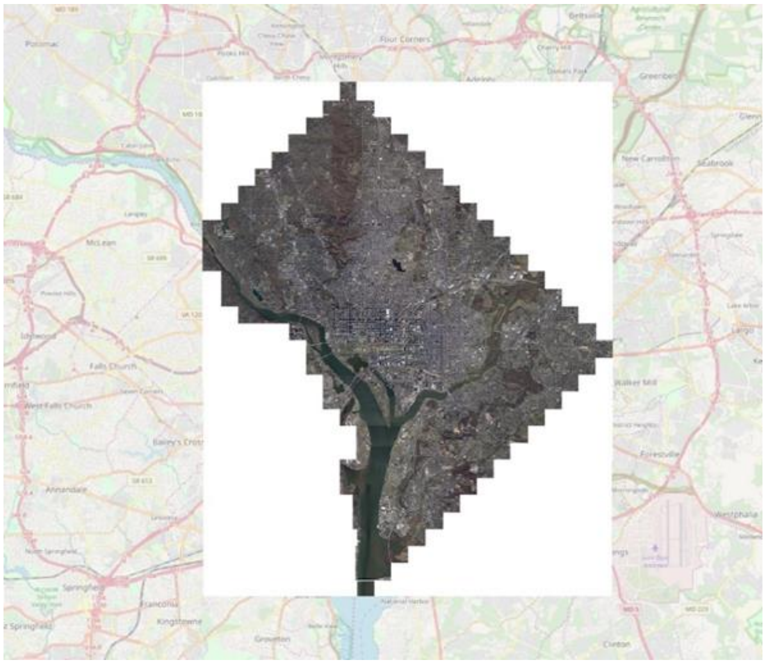


Figure 8. Visualization of Washington, D.C. aerial imagery

Similarly, a publicly available dataset was also found for Vermont[6]. Some parts of the state were available at a 15 cm resolution as represented in Figure 9; the chosen portion is located around the Essex Junction. The dimension of each tile is 900m by 900m, resulting in 36 million pixels per tile. Each dataset used featured a different resolution and tile dimensions requiring the need for a preprocessing effort to harmonize the datasets. The Vermont dataset covers more rural areas than the Washington, D.C. set. Combining both datasets provides a larger spectrum of landscapes which is necessary to export the approach to other regions of the country.

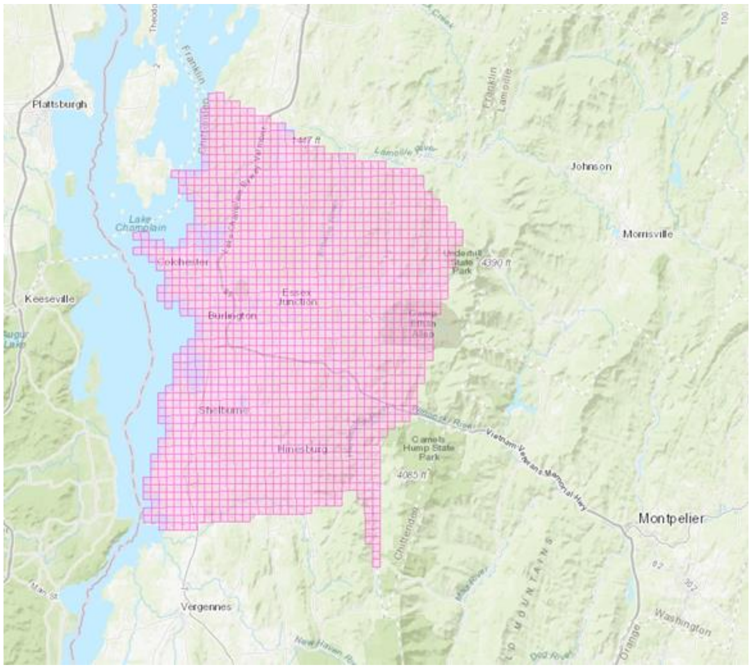


Figure 9. Portion of Vermont state available at a high resolution

Finally, the state of New York provides a publicly available set of high-resolution satellite imagery for each county (New York State GIS, 2022), (e.g., Westchester County, NY). In addition to the imagery, the locations of utility poles are also publicly available. The main advantage of this dataset was the larger area and the diversity of landscapes compared to Washington, D.C. The satellite imagery dataset covers over 500 square mile area with a 0.5 ft resolution.

2.2.3.2 Street view Imagery

The street view imagery can be downloaded by implementing a code using the Google Street View (GSV) API. The API allows downloads of about 28,000 images for free each month. Each street view image is sampled from a 360-degree camera mounted on a vehicle (i.e., car or

---

[6] https://geodata.vermont.gov/pages/imagery

bicycle). The imagery is updated every few years whenever the capturing vehicle circles through the roads.

The API works through HTTP requests. The longitude, latitude, and pitch angle of the camera can be set as input and the URL returns the corresponding image if it exists at the indicated location. For the case of wire detection, the pitch can be set as 90 degrees to angle upwards and capture images of potential wires. An example of tiles corresponding to this viewpoint is shown in Figure 10. It can be understood from this figure that the wires are much more easily identifiable using this type of imagery compared to satellite imagery.

Figure 10. Street view imagery sample taken in Westchester County, NY

2.2.3.3 Power lines and substations

Concerning the location of wire poles, the city of Washington, D.C. also publicly provides a listing of the different types of poles within the boundaries of the city and their location, in a .csv file. This dataset (Washington, D.C.) contains the location of power towers, traffic signal poles, streetlights, and utility poles, which are the data of interest. Figure 11 shows the proportion of each pole type featured in the dataset and Figure 12 shows an example image with the poles matched with the correct projections.

For the other datasets, the state of Vermont and New York published the coordinate list of the power distribution poles. Only this type of pole seems to be available in contrast with Washington, D.C. where different types of poles were included. The locations of aerial wires are also included in the Vermont dataset, which can be used to verify the accuracy of the prediction model when applied to this area.

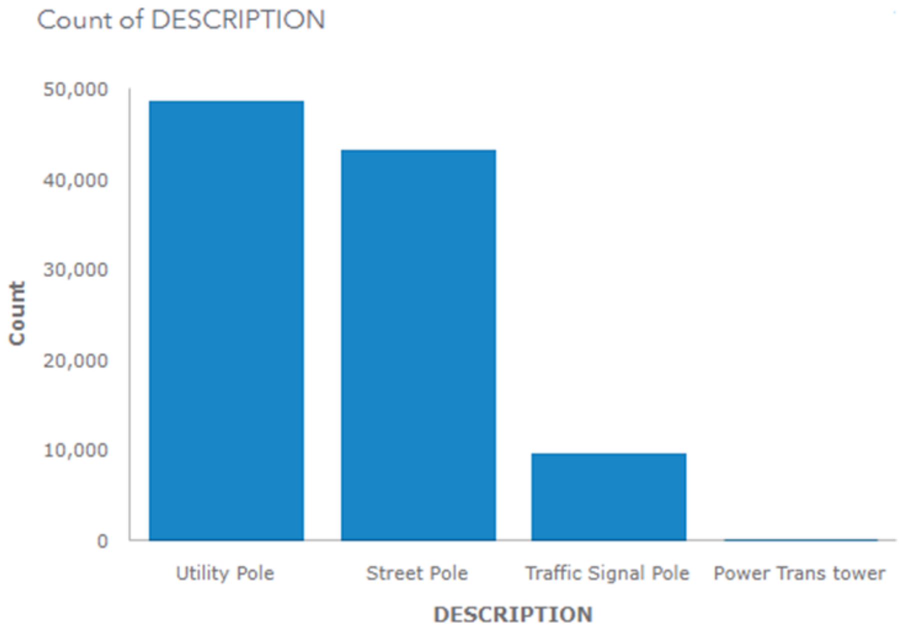


Figure 11. Pole counts in Washington, D.C. database

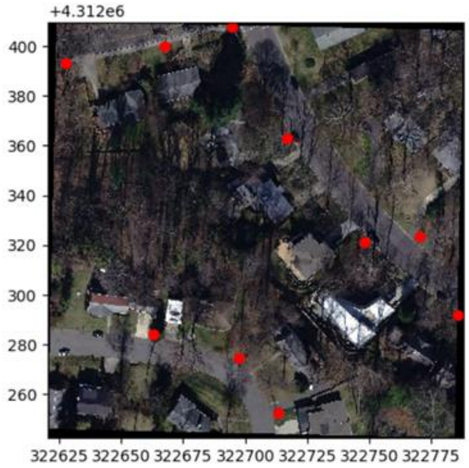


Figure 12. Pole data projected on image for Washington, D.C. dataset

2.2.3.4 Other features

The Washington, D.C. dataset, along with many other publicly available datasets, include various other geospatial features. These can include the location of roads, buildings, greenspaces, and waterbodies. An example of this is seen in Figure 13. This information was used in the initial investigations and later used in the graph search method.

Figure 13. Washington, D.C. geospatial features

2.2.4 Computational requirements

The scale of this problem causes concern for the computational requirements involved. Therefore, background research has sought to find a tradeoff between computational power and cost. An initial breakdown can be found below in Table 5, which highlights the computational power difference for each GPU that can be used to train the deep learning model. Some of the presented GPUs can be used locally from the project team member's personal laptop such as the Nvidia Quadro M1200 or using Georgia Tech cluster, the Partnership for an Advanced Computing Environment (PACE). However, those GPUs were not computationally powerful enough to perform a relatively fast training of the deep learning model. The chosen service was Google Colab Pro online which enabled the remote use of Nvidia T4 GPUs for the first phase of the data pipeline (i.e., detection) and a Nvidia GeForce RTX 2060 for the second phase of the data pipeline (i.e., prediction), as shown in Figure 6. To operate, the code needed to be uploaded on an online Jupyter notebook and could be run from any computer with internet connection.

Table 5: GPU computational power and usage pricing benchmark

GPU	CUDA Compute Capability[7]	Memory	Count	Access	Cost
Nvidia Quadro M1200	5.0	4 GB	1	Team member personal Laptop	$0
Nvidia Tesla C2050	2.0	3 GB	7 (2 nodes)	PACE (unable to access at the moment)	$0 (time-limited)
Nvidia Tesla K80	3.7	12 GB	1	Google Colab (Free)	$0 (12 hr. time-limited, many limitations)
Nvidia Tesla P100	6.0	16 GB	-	Google Colab Pro	$10 per month (1-day runtimes, some limitations)
Nvidia T4	7.5	16 GB	-		
Nvidia V100	7.0	16-32 GB	1-8	Amazon Web Services EC2 P3	$3.00-$31.00 per hour

2.2.5 Environment setup

To be able to process satellite imagery and street view imagery, train a deep learning algorithm on hundreds of gigabytes of data, and visualize the results, several different tools and a process pipeline needed to be identified. Two different dataflows were used in each phase. Figure 14 shows the process pipeline between the different tools and environments for the different steps of the pole prediction from satellite imagery for the first phase of the study. For the second phase of the study, the main differences were that the training was performed on a local machine and the result of two deep learning algorithms (one for satellite imagery and one for street view imagery) were combined.

2.2.5.1 Initial pole prediction dataflow

The initial step was the upload of two datasets. The first dataset upload was for the GPS coordinates of utility poles (stored in a .csv table) which were the target data for the ML framework. The second dataset was the actual satellite imagery stored as a list of tiles, each

---

[7] developer.nvidia.com/cuda-gpus

being a .tiff image; this was the data to predict from. Both datasets were uploaded on Google Earth Engine (GEE), which can handle large datasets and has a code editor that can perform some data processing such as the conversion of the dataset into a compressed TFRECORD file. The compressed file could then be interpreted by the ML environment, and the target data and the satellite imagery could be merged.

Once the data was uploaded, the next step was the separation of the dataset into three subsets: training, validation, and testing. The training subset is self-explanatory, this was the data used by the model for the training process to tune the network weights. Although the validation set was also used during the training, this set was used primarily to compute the accuracy of the model on unseen data, and then monitor the evolution of the performance of the model throughout the training process. Finally, the testing set was the application set. The ML environment cannot read data directly from Earth Engine; hence the different subsets were transferred to a Google Cloud Bucket which could be accessed by Google Colab, the Jupyter notebook environment where the ML algorithm environment was implemented.

The code for the deep learning implementation was written in Python on Google Colab using the Tensorflow (Abadi, 2016) and Keras Library[8]. The ML environment can directly read the datasets from the Google Cloud Bucket and transform them into a normalized matrix and divide it into training batches using the different Python libraries.

After the code implementation, the model could be trained on the Google Colab servers. The training weights were stored in the Google Cloud Bucket and the model performance evolution throughout the training was monitored using the training logs which indicated the evolution of the prediction accuracy on the training and validation data.

As soon as the training was done, the prediction model could be used on testing data. The whole pipeline was then reinitiated. First, the testing data was uploaded to GEE that converted the data into readable data for the deep learning architecture, which then used the training weights to directly output the predictions. Those predictions could then be uploaded to GEE to create a visual of the results. The outcome of the process was a list of longitudes and latitudes for each predicted pole.

---

[8] https://github.com/keras-team/keras

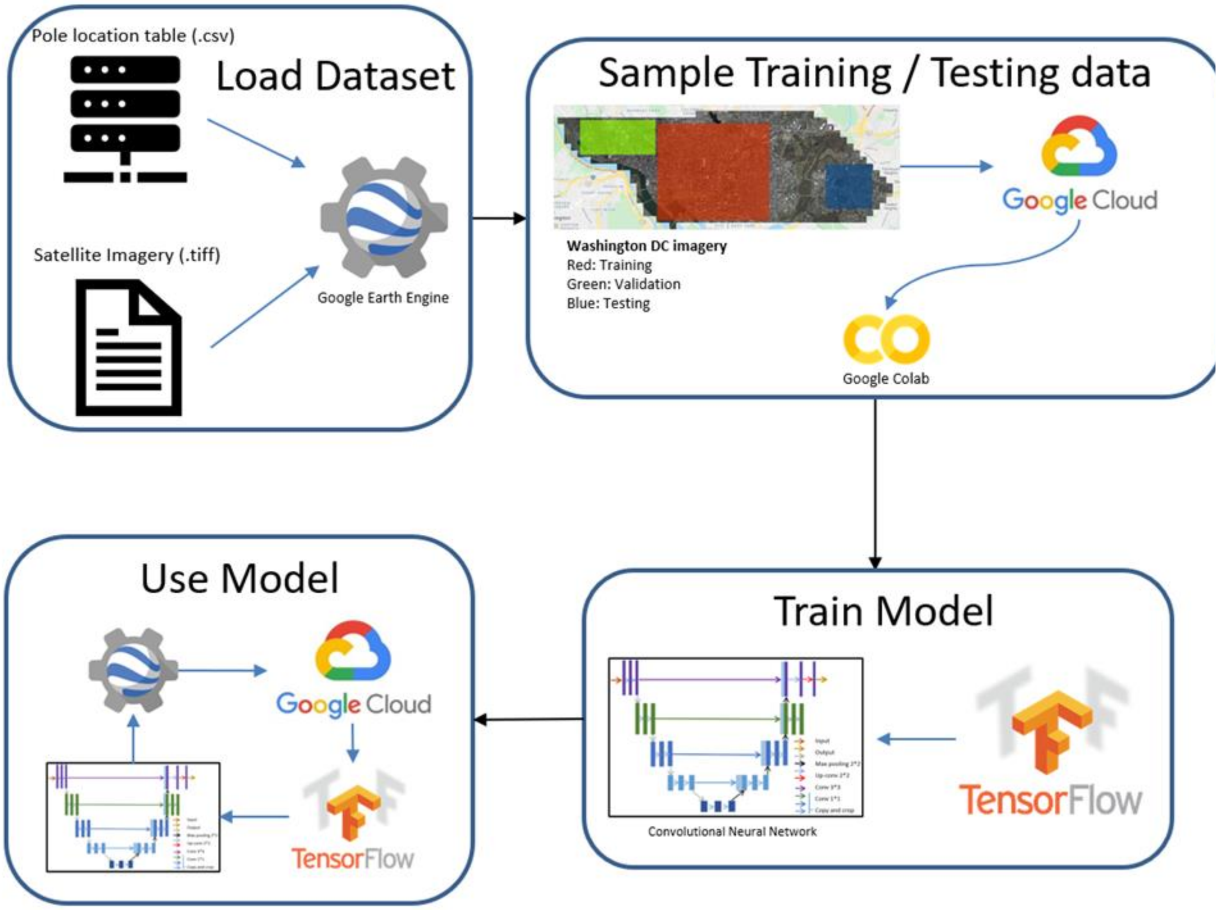


Figure 14. Pole prediction pipeline

2.2.5.2 Final pole prediction dataflow

For the second phase, the approach was similar, but the environment used was different. The online cloud services of Google Cloud and Google Colab were not used. Most of the work and training was performed locally on a GPU. As the training was local, the data flow was then different. The dataset preparations for both phases are detailed later in this document.

2.2.5.3 Data processing

As explained in the previous sections, the satellite imagery dataset was extracted from Washington, D.C. The portions used for the training, validation, and testing subsets are depicted in Figure 15. The training area corresponds to an approximate 27 square kilometer area.

Figure 15. Washington, D.C. datasets satellite imagery (training: red, validation: green, and testing: blue)

2.3 Methodology

Previous work has successfully detected wires from onboard vehicles or aerial imagery; however, the accuracy and capability were limited. The goal of this work was to form a complete database of the transmission and distribution lines; therefore, a new method was required. This work utilized deep learning for the detection of the lines or mounted poles, then a graph search method to predict the location of all undetected sections of the network. The motivation for this approach was that a two-phase process can create a higher-resolution grid network that can be used as a wire database for rotorcraft safety.

2.3.1 Remote sensing via deep learning

The field of remote sensing has shifted to using deep learning in a larger capacity, partially because of its ability to solve some of the key issues (Cira, Alcarria, Manso-Callejo, & Serradilla, 2020), such as dealing with noise, obstructions, and indistinct feature, as well as the ability of gaining insight from smaller datasets. The detection of transmission lines and poles follows this same pattern as the shape and orientations are very diverse, and it is difficult to form large training datasets.

2.3.1.1 Tile classification

Previous work by the Development Seed (The World Bank, 2018) showed the capability of using image classification to locate high-voltage towers. Tile classification is simply image classification on tiled satellite imagery. Image classification has been a critical benchmark of deep learning from the introduction of the ImageNet (Deng, et al., 2009) dataset, therefore, there is an abundance of testing and tuning that has been done with the network architectures and the training process. This leveraged itself well to get the first set of substantial results using this method.

This work was initiated with the Development Seed classification process to have a baseline prediction model. The technique involved using the Xception architecture (Chollet, 2016) . As documented by the Development Seed, the network's weights are initialized using ImageNet and a fully connected layer is added to learn the final selection for the classes in the training data. The design of the Xception network can be seen in Figure 16. One of the unique components of the network is the use of Depthwise Separable Convolutions, which combines Pointwise Convolution with Depthwise Convolutions. The training process for Xception includes parameter optimization from the *hyperopt* package.

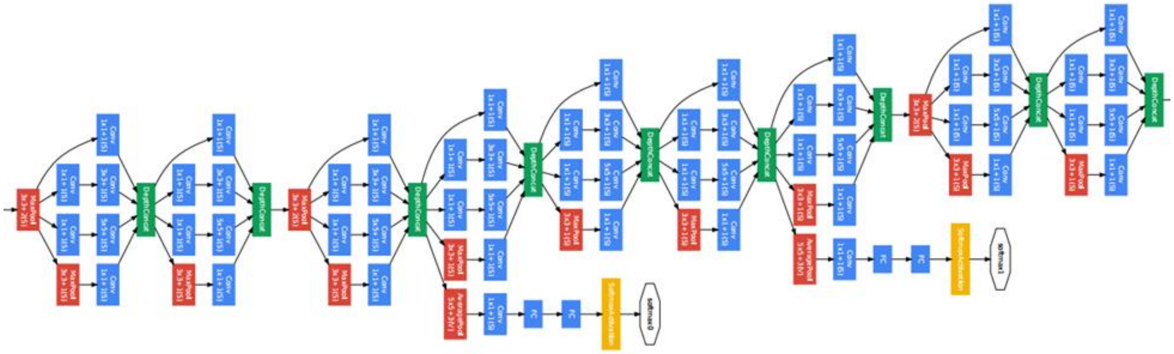


Figure 16. Xception Model

2.3.1.2 Semantic segmentation

When looking at an image, the human brain can distinguish the different objects depicted. Not only is the brain capable of listing and localizing the elements, but it can also detect the pixels corresponding to the border of each of the objects, hence performing image segmentation. In other words, semantic segmentation is the task of attributing a class to each pixel of the image. Everingham et al. (Everingham, 2012) shows an example of this process where the different classes detected are background, person, and bicycle. Recent advances in computer vision and deep learning enabled the possibility for a machine to perform semantic segmentation with the help of a convolutional neural network. Several applications are already using this technique, such as autonomous cars (Treml, 2016), detection of cancerous cells (Saha, 2018), and related to this work, satellite imagery classification (Zhou, 2018).

One of the main Convolutional Neural Network (CNN) architectures used for semantic segmentation is UNET (Ronneberger, 2015) and can be seen in Figure 17. On this diagram, each arrow corresponds to a layer operation. U-net is a symmetric network featuring an encoder and a decoder to first capture the context of the input image and then to reconstruct a prediction image.

Only convolutional layers are used in this architecture, then the network can be trained using any image size without modifying the topology of the network. To perform image segmentation, it is necessary that the dimension of the input is the same as the output of the network. While this architecture was initially implemented for biomedical image segmentation (Ronneberger, 2015), it has been successfully applied to satellite imagery segmentation problems for different image resolutions, such as impervious image segmentation (McGlinchy, 2019) or land cover classification (Garg, 2019). U-net is then a candidate approach to detect and segment wire poles from satellite imagery.

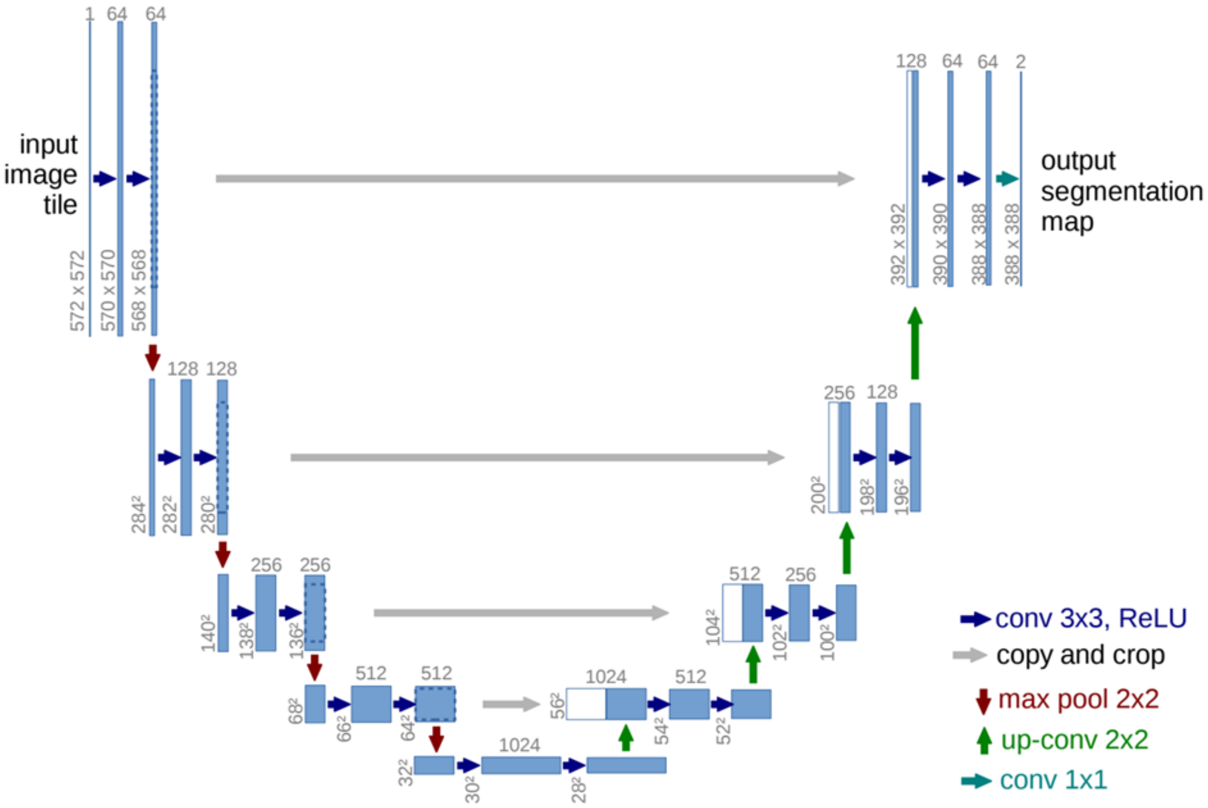


Figure 17. UNET architecture

2.3.1.3 CNN classifier

A Convolutional Neural Network (CNN) classifier was also implemented to focus on the street view imagery. This method takes a street view imagery tile as input and outputs a predicted class. CNN classifiers are one of the most used deep learning techniques for image classifications. Among all the CNN architectures presented in the literature, the chosen architecture is the Inception ResNet V2. This architecture is commonly used and can be used to classify images from the ImageNet dataset which features about 1,000 different classes (Deng, et al., 2009) indicating that this architecture is capable of handling complex images. This network is in the category of Inception architectures and is identified by the use of residual network layers enhancing the performance of the model (Szegedy, Ioffe, Vanhoucke, & Alemi, 2017). The detailed architecture of the model is displayed in Figure 18 (Alemi, 2016).

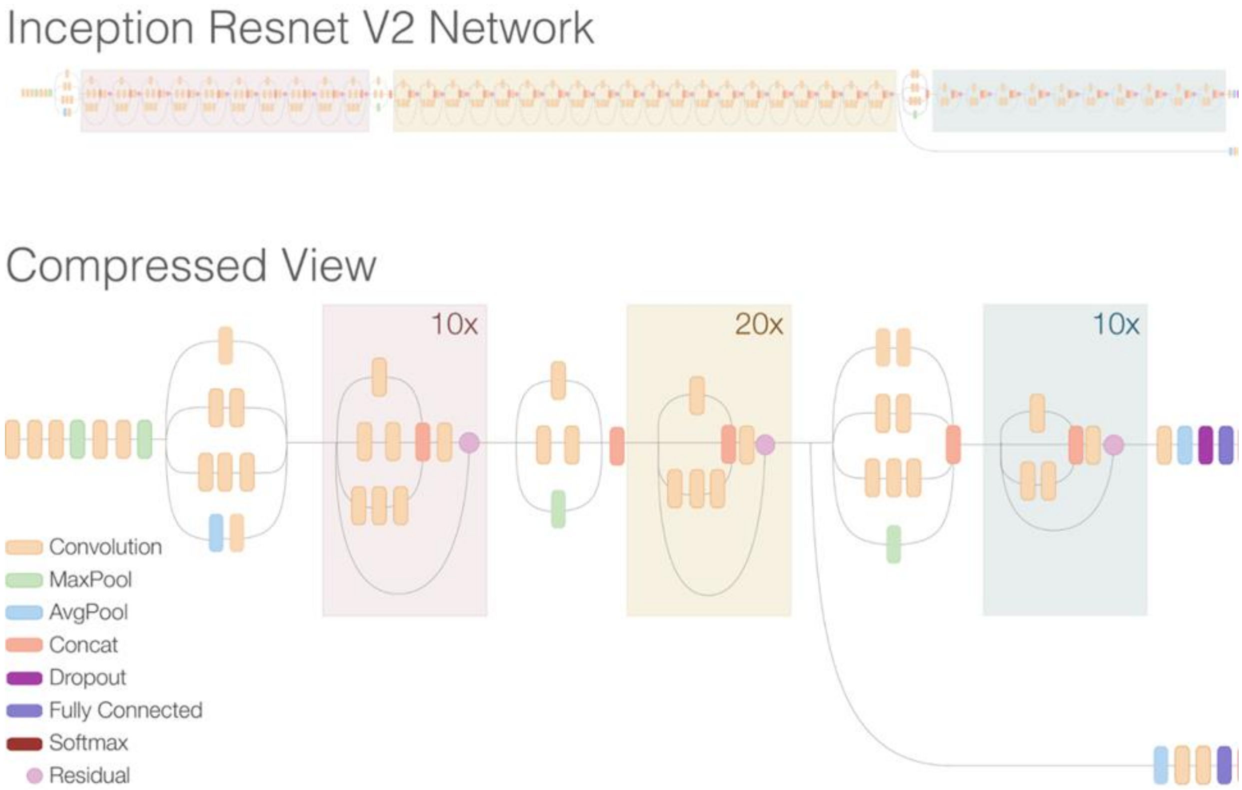


Figure 18. Inception ResNet V2 architecture

2.3.2 Grid prediction via many-to-many path planning

After the detection phase, there was still uncertainty in both where the actual wires were and in what poles were missing, from failed detections or limited processing capabilities. There are many ways to expand this information into a grid, however promising results from Facebook Engineering (Gershenson, Rohrer, & Lerner, 2019) showed that a weighted graph search using Djikstra's algorithm (Kleinberg & Tardos, 2005) could predict the grid. This work follows a process similar to Facebook Engineering; however, we used the known substations as *targets* and the detected poles as *sources*. Therefore, it was assumed that all detections are true sections of the grid that must be found in the full network topology. The centralized many-to-many version of Djikstra's algorithm, called PathFinder, expands a combined *halo* to find the best paths. The primary difference to the traditional Djikstra's algorithm is that the *halo*, implemented as a Priority Queue, is added to from all *sources* and expands simultaneously from each node in the *halo*. In this work’s implementation, it should also be mentioned that when a path is found, all sections of that path then become new *source* locations, adding to the *halo*. To bias the graph search to follow the most accurate path, a set of weights is tuned based on existing data to form the best prediction of where wires exist. Initially, this was simply the location of roads, which often is a good indicator of power lines, but additional features from openly available data include buildings and waterbodies. The assumption was that a proper selection of the weights for these features in the graph would promote the most accurate paths to form in the graph network (Figure 19).

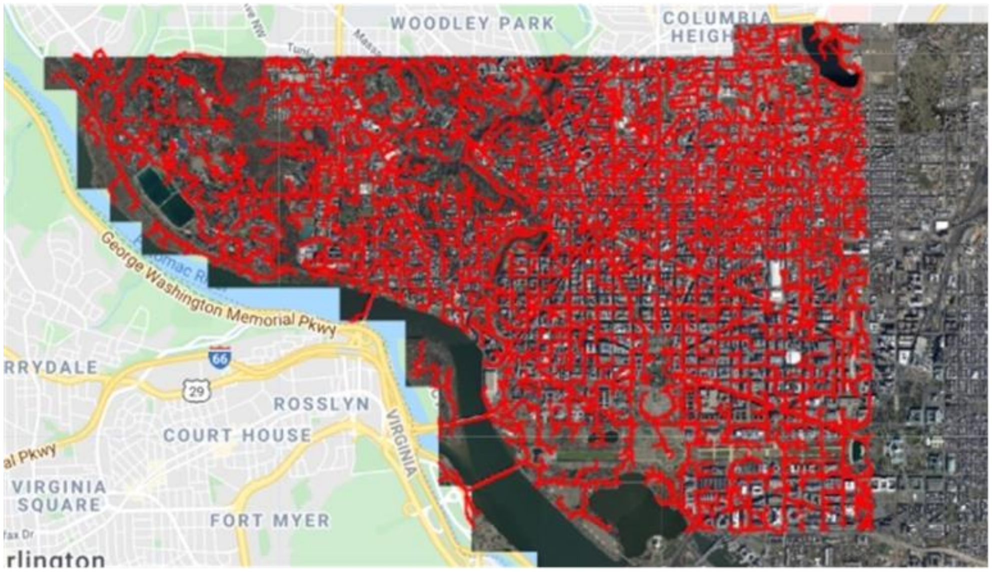


Figure 19. Graph search visualization

2.3.3 Dataset preparation

The three deep learning architectures presented previously are fundamentally different and require different input dimensions. Hence, the data needed to be prepossessed differently. The following three subsections explain the dataset preparation for the three approaches.

2.3.3.1 Xception – Tile classification

The Xception network was set to predict whether there is a pole in the image or not. The inputs for training needed to be an image and a label, and only an image for testing. The images are 256x256 pixel tiles and are normalized between -1 and 1.

Geospatial analysis on the locations of the images was needed to create a truth dataset with labels for the images. For each image, a polygon was created with the projected corner points.

Initial results from the Xception network relied on choosing mostly *no pole* classifications, partially because the dataset was oversampled. Therefore, a subset of training data removed some of the oversampled training data and in addition, the weights for the *pole* classification were increased to bias the loss function with the *pole* choices.

2.3.3.2 U-net Semantic segmentation

As explained previously, the U-net architecture takes as an input an image featuring different bands, and outputs a single band image corresponding to the feature to predict. It is then necessary to convert the satellite imagery and the pole locations to a single image. However, the pole dataset only contained the pinpoint locations of each pole. It would not be realistic to associate the location of a pole to a single pixel. The network needed to recognize the pixels corresponding to a pole; hence, more than one pixel needed to be labeled as a pole to teach the network. The ideal situation would be to have (for each pole in the dataset) the list of pixels from the satellite imagery that represent the pole. In other words, the ideal target dataset would be a segmented image with all the delimited poles. Nonetheless, this process needs to be done manually because given the size of the dataset and proportion of the data, it would be unrealistic to manually segment each pole one by one, see Figure 11. An alternative solution was found using Google Earth Engine (GEE). On GEE, it was possible to upload the satellite imagery individually using a .GEOTIFF file format, and the dataset of the pole coordinates using a .csv table. Once the dataset was uploaded, it was then possible to create a function that iterated through each pole location in the dataset, created a circle of a given radius, and converted it to an image. A 10-meter radius circle captures most of the pixels representing a pole but also labels as *pole* a significant amount of pixels around the pole. A 5-meter radius limits the number of pixels wrongly labeled but does not capture all of the pixels pertaining to a pole. Figure 20 shows a comparison of labeling using 10-meter radius circles and 5-meter radius circles. For this case, the authors decided to use a 5-meter radius labelling to minimize the number of pixels around each pole which can confuse the deep learning model. In this dataset, only the utility poles are of interest, yet if the target image was using only the coordinates of the utility poles, the image would be too sparse and most of the pixels would be labeled as *no pole* and as experienced by the authors, the model would settle to predict only *no pole* pixels but still have good accuracy to predict true negatives, which would be the majority of the data. To counter this problem, the other types of poles were also included in the target image to reduce the imbalance of pixels labelled as *pole* or *no pole*. The target image then contained a single band with a different pixel value for each pixel depending on the type of pole represented. Figure 21 shows the pixel value used for each pole and illustrates a fraction of the image produced using this label code. The target image was then concatenated to the satellite imagery on GEE creating an image featuring four bands, the last one being the target, the pole value.

The last step of the dataset preparation consisted of sampling the image choosing the area corresponding to the training set, a different one for the validation set and a last one for the testing dataset. The approach was different for the first and the second phases as the environments used were different.

2.3.3.2.1 Dataset preparation for first phase of study

This next steps after creating the label image consisted of converting and transferring the data to the U-net model for the training. The training and evaluation images were converted to a .tft record file and transferred to a Google Cloud Storage (GCS) bucket. The U-net model was then implemented using python and TensorFlow on Google Colab which has an integrated pipeline with GCS and GEE. After the training of the model, the testing set was uploaded to GCS and plugged into the trained model to generate the predictions which were uploaded to GEE to visualize and analyze the results.

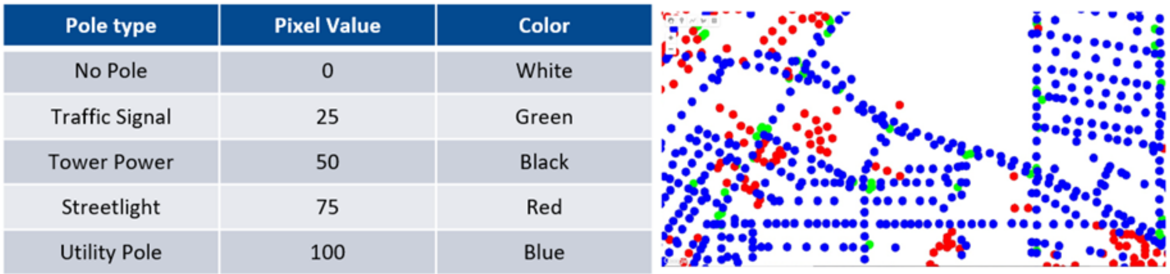
2.3.3.2.2 Dataset preparation for second phase of study

For the second phase, as the models were trained locally, the dataset preparation was different. The two datasets used for this phase were Vermont and Westchester. The Vermont dataset provided a set of satellite imagery stored in a .tiff format. Westchester County formatted the imagery in a proprietary file extension .sid. The files were converted to a .tiff format using a Python script. Once both datasets were in a .tiff file format, each tile was sampled into smaller tiles of 250 pixels by 250 pixels to reduce the computational cost. In addition to this process, the small tiles were filtered to remove the tiles that did not contain any wires. As both datasets had rural landscapes contrary to Washington DC, the imbalance between the data labeled as *no pole* and *pole* was reduced.

2.3.3.3 Inception ResNet V2 classifier

To download street view imagery tiles from the GSV API, HTTP requests need to be called using the longitude, latitude, API key, and camera pitch angle as input. To be able to view wires, the camera pitch angle needs to be set as 90 degrees corresponding to an upward aim. To generate a dataset, the areas of Westchester County and Vermont were divided into a thin grid of longitude and latitudes. Once the grid was generated, each grid point was tested through an HTTP request to retrieve the meta data of the imagery tile. If the metadata exists, the tile was downloaded. This method generated approximately 20,000 street view images for each dataset. However, unlike the UNET approach, the ground truth for the dataset of Westchester was unknown. The GSV approach aims to predict the location of wires while UNET only focuses on utility poles. At the time this study was conducted, the county of Westchester did not have a wire location dataset. Hence, labeling of the dataset needed to be done manually. As the GSV approach uses a CNN classifier, the labeling phase consisted of copying predicted images into two separate folders corresponding to either: *contains wires* (1) or *does not contain wires* (2).

Figure 20. Comparison of pole trust region radius



Pole type	Pixel Value	Color
No Pole	0	White
Traffic Signal	25	Green
Tower Power	50	Black
Streetlight	75	Red
Utility Pole	100	Blue

Figure 21. Labeling of pole dataset

2.3.4 Grid prediction pipeline

The grid prediction required both dataset pre-processing to make predictions, and pole detection post-processing for the full pipeline. First, the grid must be created based on the region of interest. This required the polygon of the region, the projection of the geometry, and the desired grid square size. The polygon was projected to a rectangle to make a simple subdivision across the shape. If necessary, the grid outside the boundary can be removed, although for convex polygons, if all the targets and sources are within the boundary polygon then there should not be any change in the results or convergence time. Next, the feature datasets were provided in vector form from either a publicly available dataset, such as Open DC, or from OpenStreetMap. These features must be rasterized into the grid's spatial resolution to be used as weights. This was done using the package GeoPandas, which utilizes R-tree spatial indexing. Although, since the data covers a majority of the regions in the boundary, this had limited performance improvements. Therefore, the region is sub-divided beforehand to give an additional boost to the spatial index queries.

The extra step of subdividing the region required the grid prediction to be done with the full pipeline to setup the sources and targets for the many-to-many graph search.

However, the outputs from the pole detection method requires post-processing, since the semantic segmentation method outputs are pixel-wise classifications.

2.4 Results

The Washington, D.C. region was selected for the proof of concept of the detection and prediction pipeline, for the first phase of the study. In the second phase of the study, the dataset of Vermont and Westchester County were used. The main advantage of Washington, D.C. was to provide both utility pole location and high-resolution aerial imagery as open-source datasets. The other two datasets featured a more diverse landscape.

The next sections that follow show the progression of both phases (detection and prediction) using the different datasets. Then, the full pipeline's results are examined with a discussion on accuracy of the data and the visualization of the information. Lastly, some details are provided on the sensitivity of the models to the input data and the confidence of the predictions based on the fidelity and uncertainty of both the models and input data.

2.4.1 Power pole detection

In this section, the result of the power pole prediction from satellite imagery will be presented for both image classification methods: UNET and Xception. The Xception network provided a baseline, which had previously showed promising results. However, limitation of the network to the Washington, D.C. dataset led to the need for another model. The UNET model showed more promising results and is now the core model for pole detection, which is described in later sections. This comparison was performed uniquely on the Washington, D.C. dataset as part of the first phase of the study.

2.4.1.1 Preliminary investigation with Xception

The utilization of the Xception model was initiated from the framework by the Development Seed[9]. The framework provided a technique and structure to apply tile classification for utility pole detection in Washington, D.C.

An issue from the start for the Xception network was a sensitivity to the quality of the training images. This makes sense as weak signals from blurry or uncertain images, and false signals from incorrect labels will lead to poor training. Therefore, a hand-labelled subset of the data was formed using the online tool makesense.ai[10] shown in Figure 22. This tool helped to remove visual inconsistencies from the dataset, such as errors in pole location, blurry images, and undetectable poles in shadows.

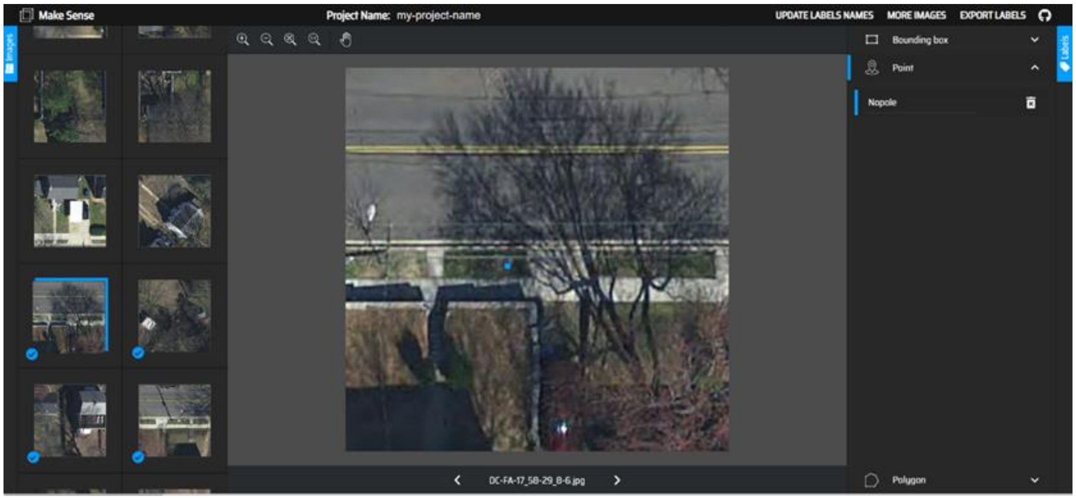


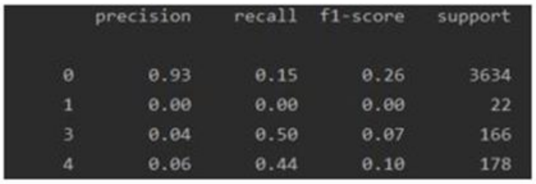
Figure 22. Makesense.ai tool for hand-labelled dataset

Now, with the improved dataset the Xception network was trained. The dataset was split 70%, 15%, 15% for training, validation, and testing respectively. This resulted in around 600 *pole* classified images and 3,000 *no-pole* classifications. When applied on the full dataset, the model achieved the results shown in Figure 23. It can be seen that the accuracy to predict no poles is very good but extremely poor to predict any type of pole. On the samples displayed, the graphs show the predicted probability for each class; the highest probability represents the predicted class. A false prediction is highlighted in red; it is then clear that the Xception architecture fails to predict any type of pole.

---

[9] https://github.com/developmentseed/ml-hv-grid-pub

[10] https://www.makesense.ai/



```
              precision    recall  f1-score   support

           0       0.93      0.15      0.26      3634
           1       0.00      0.00      0.00        22
           3       0.04      0.50      0.07       166
           4       0.06      0.44      0.10       178
```

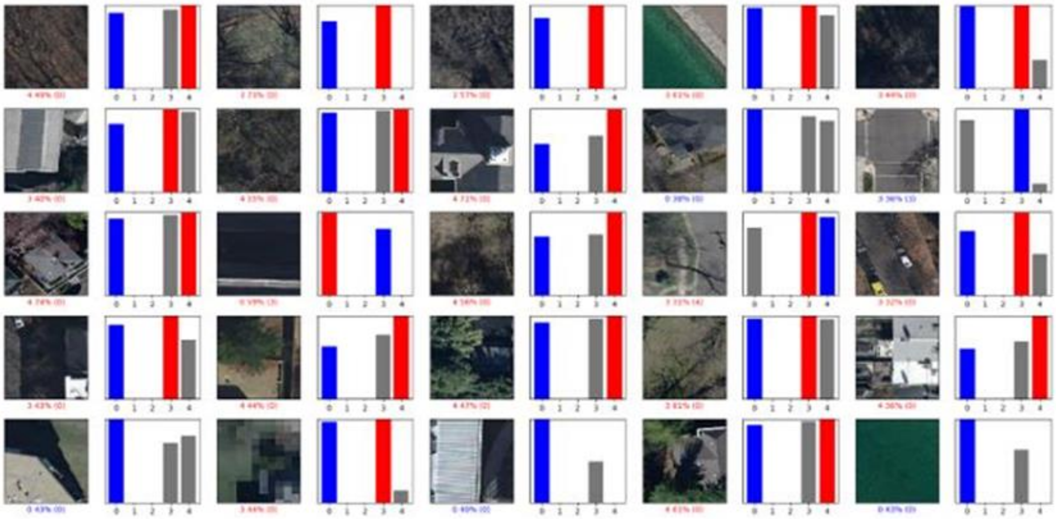


Figure 23. Preliminary results from Xception Network with compiled dataset

Despite the use of hyperparameter optimization methods, training Xception using a Google Colab Pro account was challenging. Figure 16 shows that there are many weights to compute for the network, and the dataset of 70,000 images of 256 by 256 pixel each makes the training very computationally expensive. The hyperopt process multiplies the time required for training the model by the amount of optimization iteration. The runtime limitation of Google Colab forced the team to train the model by batch, 8 hours a day for a total of 35 epochs in a week. The results achieved with the Xception network training were not satisfying. For those reasons, the team decided to place the focus on UNET for the rest of the study.

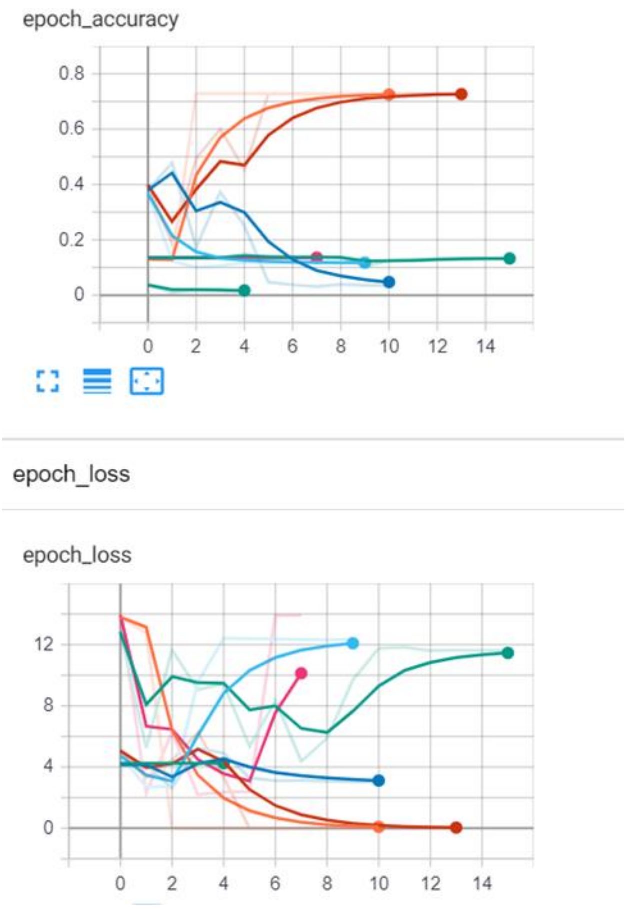


Figure 24. Loss function evolution during the model training

2.4.1.2 Successful detections with UNET model

The most promising approach investigated so far was the application of the UNET architecture. The model had been trained continuously for a week using the Google Colab server on the Washington, D.C. region highlighted in red on Figure 15. The model was first applied on the testing region highlighted in blue. The ground truth data of this region was used to measure the accuracy of the model. The training accuracy and loss evolution is shown in Figure 24.

2.4.1.2.1 Visualization of the prediction image

As the output of the UNET architecture is an image, some data post-processing is necessary to be able to assess the performance of the model. Figure 25 shows a portion of the raw predictions applied on the testing regions and Figure 26 shows the ground truth for the same portion. The pole pattern is similar, but a difference of color for predicted poles can also be noticed. It appears that the UNET model struggles to identify the difference between the various types of poles. As the objective is to mainly use the model in remote and rural regions, hence mostly non-urban areas, predicting any sort of pole can be acceptable. Yet, this shows the limitation of this architecture and highlights the need for a larger training area, and possibly a benchmark with other prediction methods in future research.

The prediction image is subtracted to the ground truth image. This process enables the possibility to have a visual of the correctly predicted pixels and the mislabeled ones. For each pixel, there are four possibilities. If the pixel is predicted as *pole* and the prediction is correct, the pixel is then a true positive, otherwise it is a false positive. If the pixel is labeled as no *pole* and the prediction is correct, the pixel is a true negative, on the contrary if the prediction is inaccurate then the pixel is a false negative. In other words, the higher the proportion of true positives and true negatives, the better. Figure 27 shows a portion of the prediction results for the testing region after being subtracted to the ground truth.

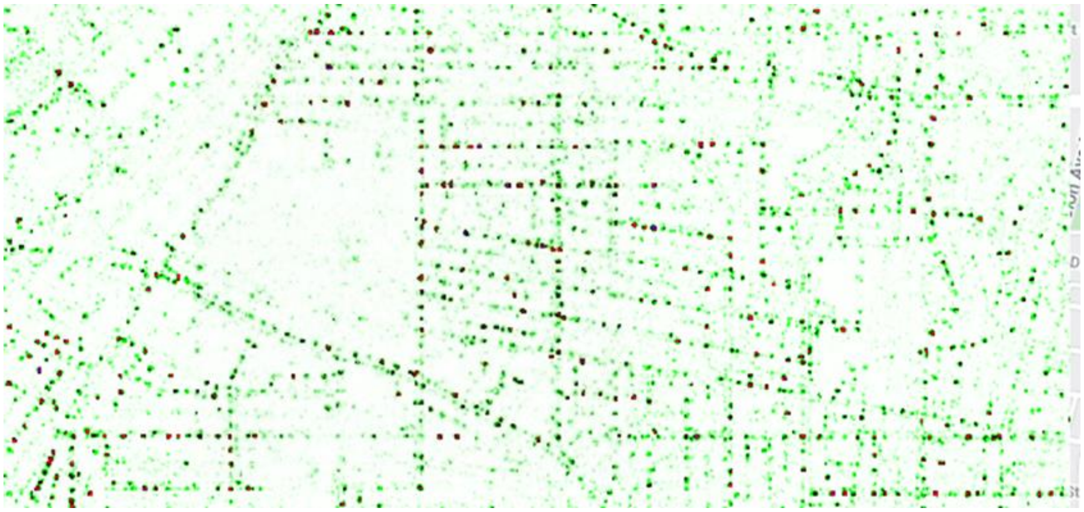


Figure 25. UNET raw predictions on testing region

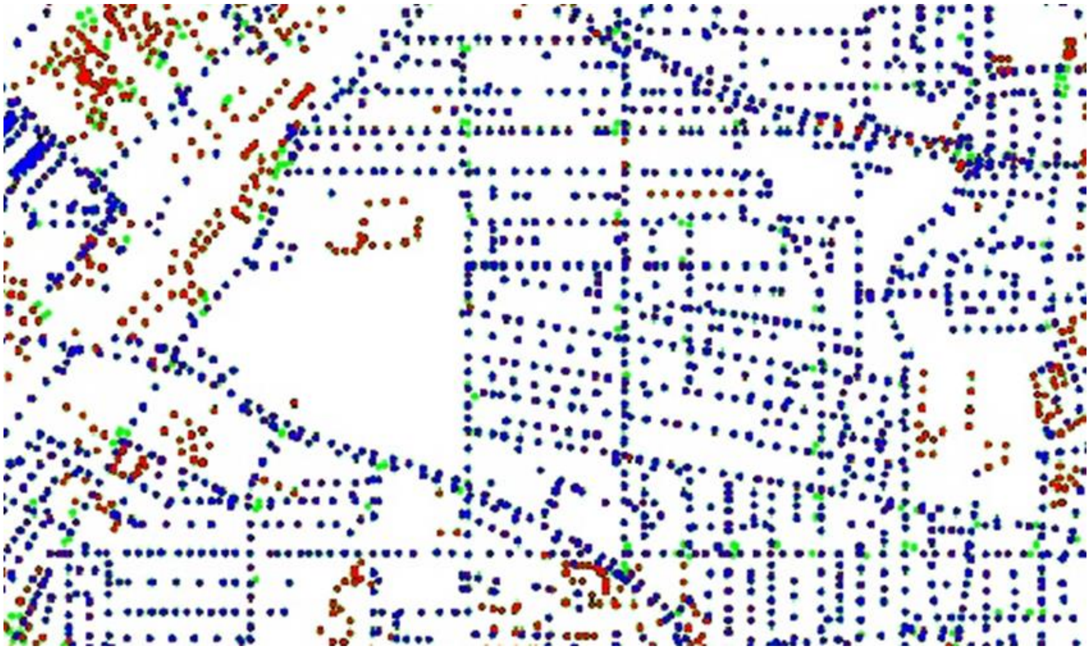


Figure 26. Ground truth image on testing region

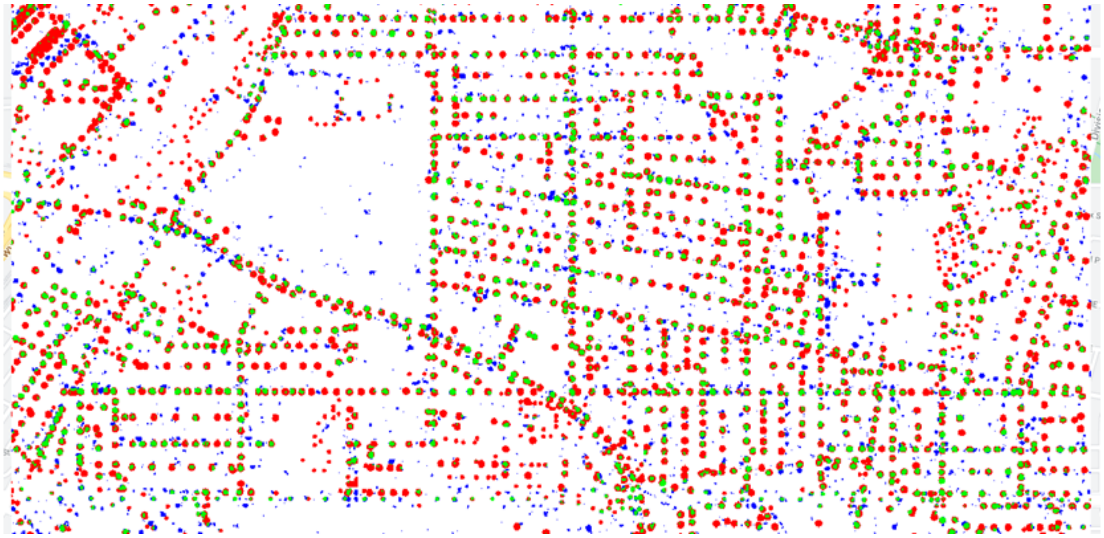


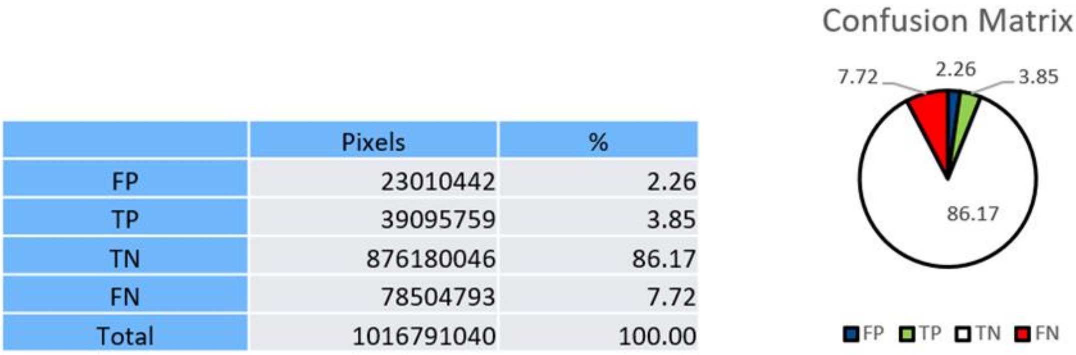
Figure 27. Portion of UNET prediction result (white: true negative, green: true positive, red: false negative, blue: false positive)

2.4.1.2.2 Model performance assessment

With Figure 27 it is possible to precisely assess the performance, hence the accuracy of the model by counting the proportions of true and false negatives with respect to true and false positives. This process can be performed on GEE and Figure 28 shows the confusion matrix of the results. From those percentages, a large variety of metrics can be computed, for example the Matthews correlation coefficient (MCC) detailed in Equation 1.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \qquad 1$$

Equation1 outputs a number between -1 and 1 and rates the quality of the predictions. If the MCC is close to -1, the predictions are the opposite of the expected predictions, if it is close to 0, the prediction is rated as random and finally, 1 is the perfect prediction. After training the deep learning model, the highest MCC achieved was 0.41. This number should be higher since the model does not predict a perfect circle for the trust region for each pole. This means that even if a pole can be correctly labeled in the center of each labeled pole circle, any pixels in the border wrongly labeled due to the imperfection of the predicted circle, will be labeled as false negatives or false positives. To illustrate this, Figure 29 shows a zoomed-in prediction image. On this figure, it can be noticed that every pole is correctly predicted since each pole circle center is painted as green. As the objective is to obtain pinpoint locations for each predicted pole, the red color at the border of each circle does not indicate an imperfection in the predictions, yet these pixels were included in the counts for false negatives.



	Pixels	%
FP	23010442	2.26
TP	39095759	3.85
TN	876180046	86.17
FN	78504793	7.72
Total	1016791040	100.00

Figure 28. Confusion matrix of the UNET predictions on the testing area

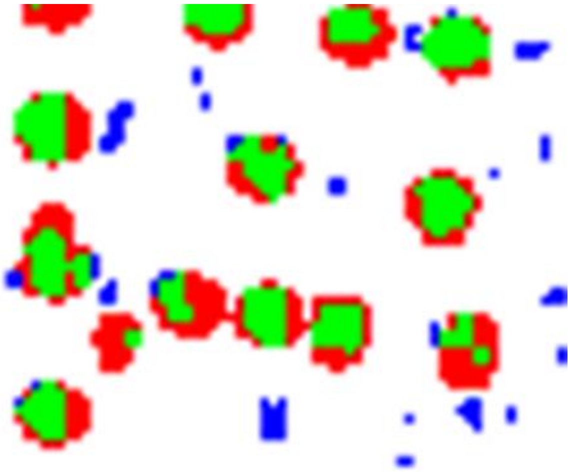


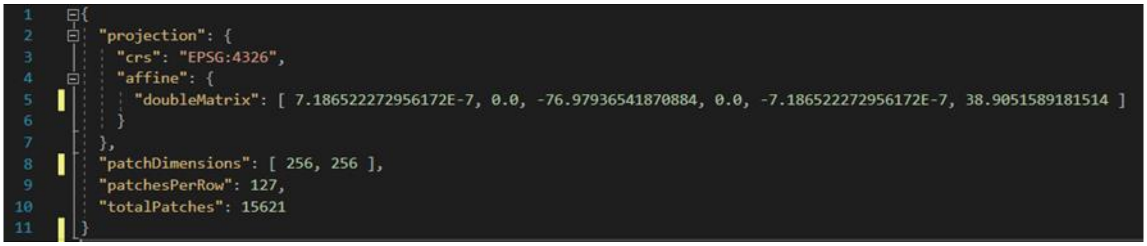
Figure 29. Zoomed-in prediction result (white: true negative, green: true positive, red: false negative, blue: false positive)

2.4.1.2.3 Longitude and latitude extraction for the predicted poles with UNET

Once the UNET architecture outputs an image, additional processing is necessary to extract the longitude and latitude of the predicted poles to then apply the grid prediction model to generate a predicted wire map. The first approach was to perform a filtering process and the longitude and latitude extraction on GEE. As there are millions of pixels to process, this method can be time consuming. As an example, the testing region took about a day to extract the coordinates of each predicted pole. As the ambition is to expand this approach to larger areas such as an entire state, this process would not be the most optimal. An alternative process was found and will be detailed in this section.

Right outside the network, the prediction output is a vector of patches of 256 by 256 matrices. Those patches can be reshaped into a single matrix. Using mathematical operations, the dimension of this matrix can be reduced, taking the average of portions of the original matrix. Then, every element of the reduced matrix is analyzed and extracted if the element corresponds to a pole. Finally, using the matrix coordinates and the meta data of the predictions, the longitude and latitude of the predicted pole can be retrieved and stored in a table. Figure 30 provides an example of the meta data of the prediction image, stored as a JSON object, that can enable this processing illustrated in Figure 31. The important values are the number of patches per row which can enable the possibility to reshape the vector of patches into a matrix, and the double matrix that provides the linear coefficient to convert the matrix indexes into longitude and latitude.

The outcome of this process is a list of longitude and latitude coordinates of the predicted poles. A shape file grouping all the pinpoint coordinates can be created and visualized as shown in Figure 32.



```
{
  "projection": {
    "crs": "EPSG:4326",
    "affine": {
      "doubleMatrix": [ 7.186522272956172E-7, 0.0, -76.97936541870884, 0.0, -7.186522272956172E-7, 38.9051589181514 ]
    }
  },
  "patchDimensions": [ 256, 256 ],
  "patchesPerRow": 127,
  "totalPatches": 15621
}
```

Figure 30. Example of JSON object storing the meta data of the prediction image

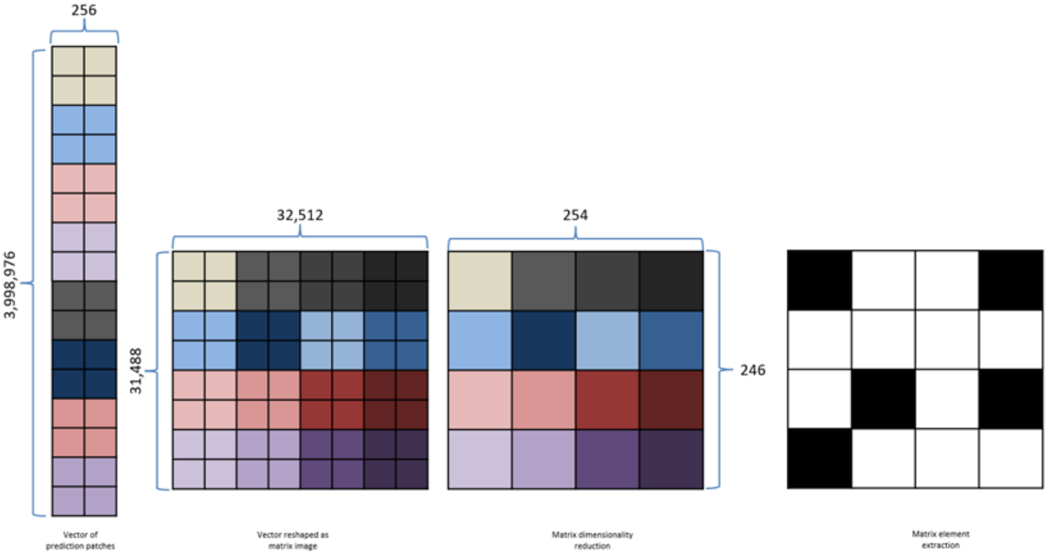


Figure 31. Example of pole coordinate extraction from image matrix

Figure 32. Example of pole coordinates extraction result

2.4.1.2.4 Unsatisfactory results using the Vermont dataset

A first attempt to train both models was done on the Vermont dataset. The Google Street View model was applied first to a portion of the dataset. The raw results are illustrated in Figure 33. Each dot corresponds to a predicted wire location. As the results were captured from a street, each prediction point was projected to the closest road using a K-nearest neighbor approach. Most of these images were captured using a 360-degree camera mounted on a car or a bike, hence it is safe to assume that each image sampled corresponds to a location alongside a road. Using a plugin from the QGIS software (QGIS Development Team, 2022), it is feasible to project a set of points to a set of lines. Each predicted point was then projected to the closest road. The result of this process is shown in Figure 34. As all the prediction points are along roads, taking the intersection of the point locations with the streets generates the wire map for the region. The wire map for the portion of Vermont is shown in Figure 35. The accuracy for this pipeline is high, proving that the GSV approach can identify wires alongside roads with a high accuracy.

Figure 33. Raw results of GSV model applied in Vermont (zoomed in view at bottom)

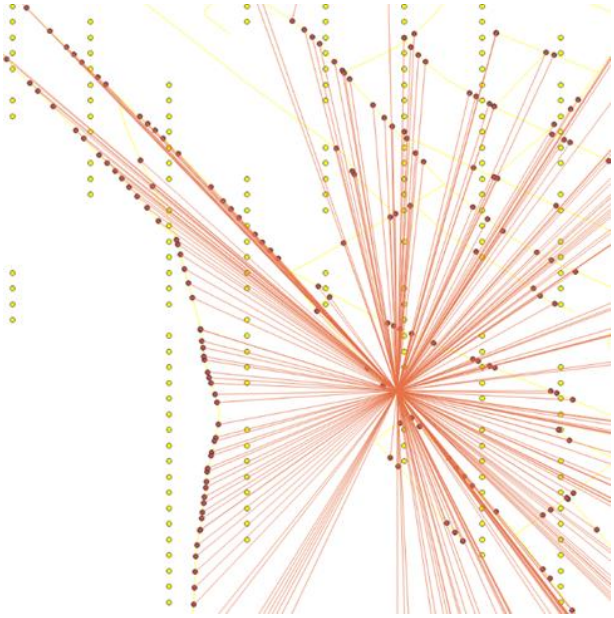


Figure 34. K-nearest neighbor approach to project GSV predictions to closest roads

Figure 35. Wire map of the portion of Vermont from the GSV pipeline

Concerning the satellite imagery pipeline, training attempts provided unsatisfactory results with the UNET model which focused on predicting *no pole* for any tile. As shown in Figure 36, the pole location dataset from the state of Vermont can be confusing for the model. There seems to be missing data and some poles are obstructed by trees making it challenging for the UNET model to distinguish poles from the rest of the image. As the result from the GSV cannot sufficiently provide a complete wire map for this data, the team decided to move on to a different dataset, the Westchester County.

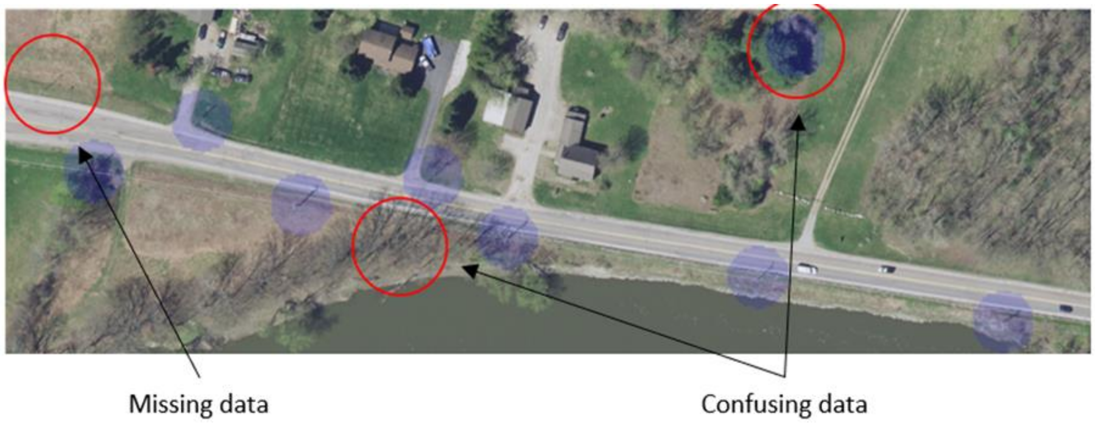


Figure 36. Vermont incomplete pole dataset

2.4.1.2.5 Results using the Westchester County (UNET model)

Initially, the satellite imagery dataset had different types of poles as labels. There were utility poles, streetlights, traffic signal poles and power towers for Washington, D.C. Yet, for Westchester, only a list of utility pole locations was published. Even though it is the only type of pole that is relevant to the prediction of aerial wires, the consequence of this difference is the low proportion of pole-related pixels compared to non-pole pixels. The more type of poles included in the label mask, the fewer non-pole pixels there will be. This imbalance of labels in the image will reinforce the behavior of predicting *no pole* as the model would be correct most times. This also explains the difficulties from the Vermont dataset. In the first phase of the study, the UNET model had difficulties differentiating between the different types of poles as they look identical when looking from above. Having only utility poles in the dataset can help the model to focus on detecting only one type of pole teaching it the differences with the other types. This means that for this dataset, the material is less dense but has more information.

For the UNET model, the satellite imagery tiles were split randomly into a training and a testing dataset. Similar to Vermont and Washington, D.C., the model was prepared with the training set and then applied to the testing images. Using the same technique as in the first phase of the study, the predicted pixels were converted into a prediction location by applying a grid and filtering the results, shown in Figure 37. The outcome is a list of pole location predictions.

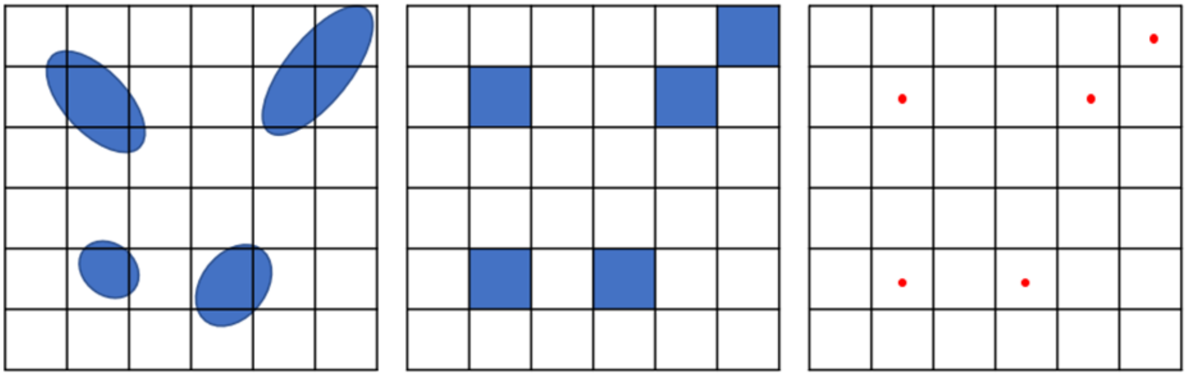


Figure 37. Filtering process and extraction of pole coordinates from the U-Net model

The prediction from the UNET model on the testing portion is shown in Figure 38. Each pole predicted is illustrated as an orange circle having the outcome point location of the filtering process at its center. On this image, the radius of the prediction point circles has been exaggerated to illustrate the uncertainty of the model prediction using the UNET approach. The actual locations of the poles are also shown on the right side of the figure. There was a small number of predictions compared to the number of poles in the truth dataset. This imbalance has been explained in the beginning of the section, the UNET model struggles to identify poles when there are less pixels corresponding to a pole in the training dataset. However, when analyzing the results, most predicted poles by UNET were correct. This lack of prediction establishes the need of combining this approach with the street view imagery model to generate a more complete wire map.

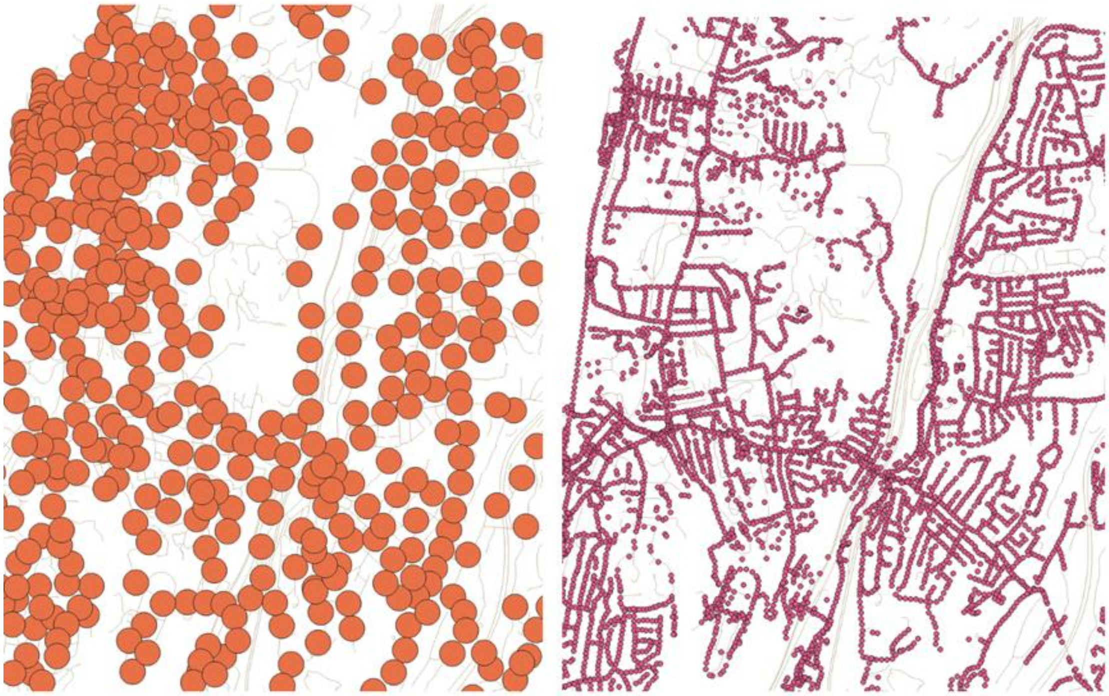


Figure 38. UNET model prediction results (testing set: left, ground truth: right)

2.4.1.2.6 Results using the Westchester County (GSV model)

Street view imagery tiles were extracted for Westchester County. The CNN classifier did not need to be retrained as it was successfully trained and applied to the Vermont region. Even if both regions have different landscapes, the upward street view imagery looks identical. Once the model was applied to this dataset, only the tiles labeled as *contains wires* were kept, and the longitude and latitude of the image captures were retrieved from the meta data of the tile. From this, a list of GPS coordinates corresponding to predicted locations was generated. A fraction of the raw output of this process is displayed on the left side of Figure 39.

Figure 39. Raw predictions of Google Street View approach (left) projected on closest streets (right)

The raw results, displayed in Figure 39, were projected to the closest roads using the same framework as for the Vermont dataset, the K-nearest neighbor approach. The result of the projection is shown in Figure 39 on the right side. The Westchester County also publicly provides a dataset of the center lines of each road as a line shapefile which made the projection possible.

2.4.1.3 Result combination

Comparing the results of UNET and GSV, a combination of the two models makes sense. The UNET model can predict poles, yet not as many as the CNN classifier approach. On the other hand, the street view model has a greater accuracy for wires that are alongside roads but cannot predict off-road wires as no imagery is available in such configuration. Moreover, the street view imagery approach directly predicts wire positions while the UNET model solely predicts pole locations. This difference can also explain the difference in terms of the number of predictions provided by both approaches. A union of both point clouds makes sense. A wire is connected to a pole, hence the pole predictions from UNET can be interpreted as a wire location prediction as well. The result of the union of both models is a larger point cloud of predicted wire locations. The outcome of the merge is shown in Figure 40. This union was performed using the data manipulation tools of the QGIS software.

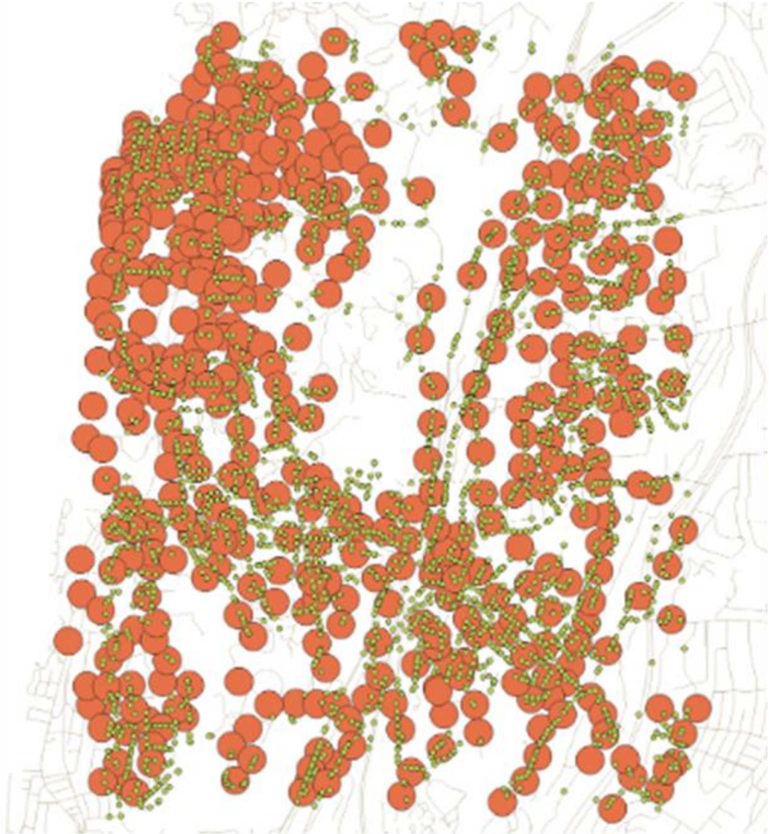


Figure 40. Combination of results generated by UNET (orange) Inception Resnet V2 (green)

2.4.2 Power grid prediction

The network prediction uses a many-to-many Dijkstra algorithm to trace paths from substations to pole detections. The detections from the previous section were used as the source points to expand to the target substations. The grid weights are a combination of feature *weights* and *costs*, which are described in an earlier section. The results of the grid prediction are discussed in the following sections.

2.4.2.1 Preliminary results

The first set of results was for a subset of the Washington, D.C. region. The flexibility in the implementation allows any region to be tested if a boundary is provided with the proper geospatial transformation information. This is often compressed in a geojson (.geojson) or shapefile (.shp). The only features used were roads in the Washington, D.C. area, which were weighted at 10% cost of the other nodes. Therefore, a 10-node path along a road is the same cost as moving through any other single node.

The test case is for a region of Washington, D.C. where 21,534 pole detections were made and 10 substations exist. The grid cell size was 49 m^2 and the algorithm took approximately 19 minutes for 1.78 million iterations. Some sample results are shown in Figure 41, Figure 42, and Figure 43 such as:

- Number of grid cells in region of interest: 1,788,720
- Area of grid cells in region of interest: 49
- Number of origins in region of interest: 21,534
- Number of targets in region of interest: 10
- Number of Pathfinder iterations: 3,178,283
- Pathfinder completion time: 1141.24 s

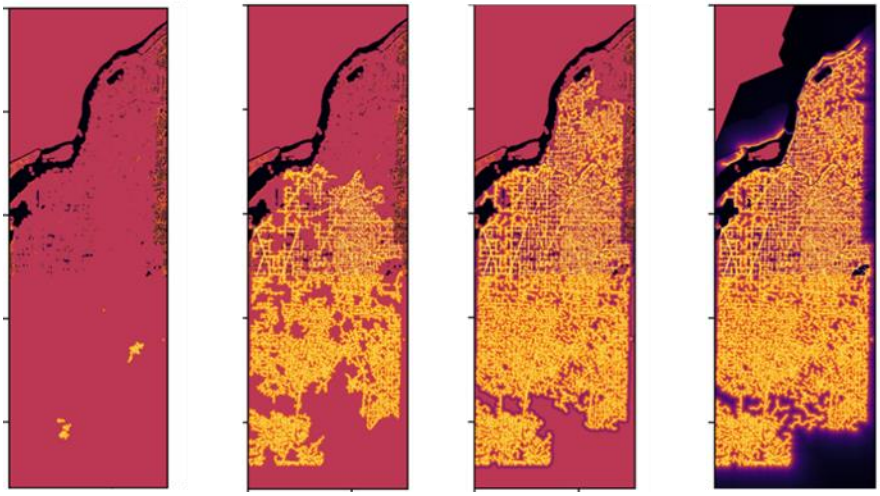


Figure 41. Graph search map of costs for increasing iterations (Washington, D.C. test case)

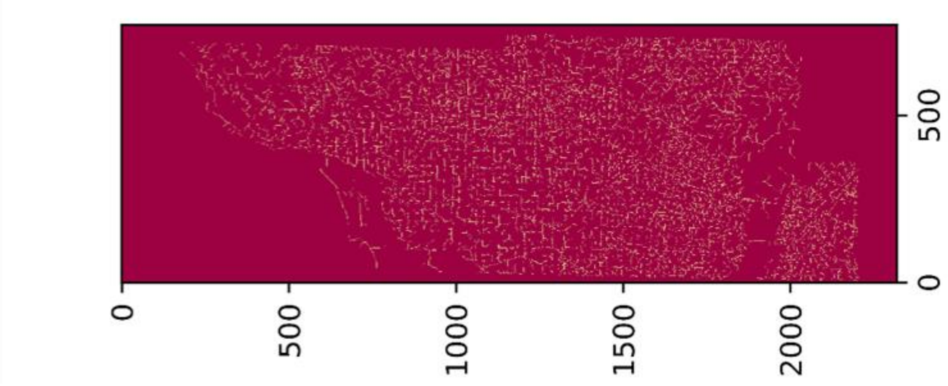


Figure 42. Final path results for Washington, D.C. test case

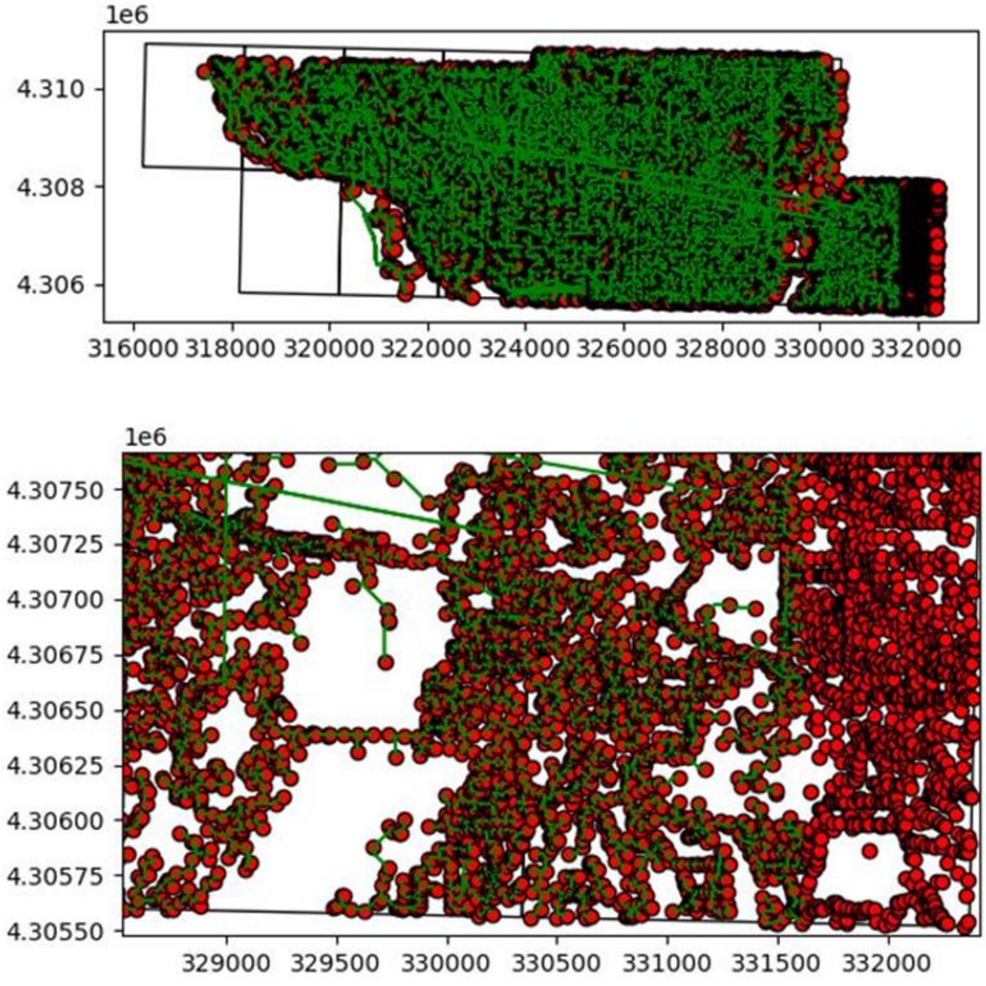


Figure 43. Visualization of path (Washington, D.C.)

2.4.2.2 Generating "Truth"

For this step, the objective was to create a truth wire dataset in order to benchmark the generated results. For the future Vermont dataset, the power lines are given in the truth data but the D.C**.** dataset only has poles. A "truth" dataset was created using the truth pole data and the graph search grid prediction. It is assumed that the weights, since they haven't been tuned, are less important for the algorithm when there is knowledge on all poles (i.e., complete knowledge). The algorithm ran for 6 hours including 4 hours of preprocessing and 2 hours of graph search and generated a array and a shapefile containing the "true" wire map displayed in Figure 44.

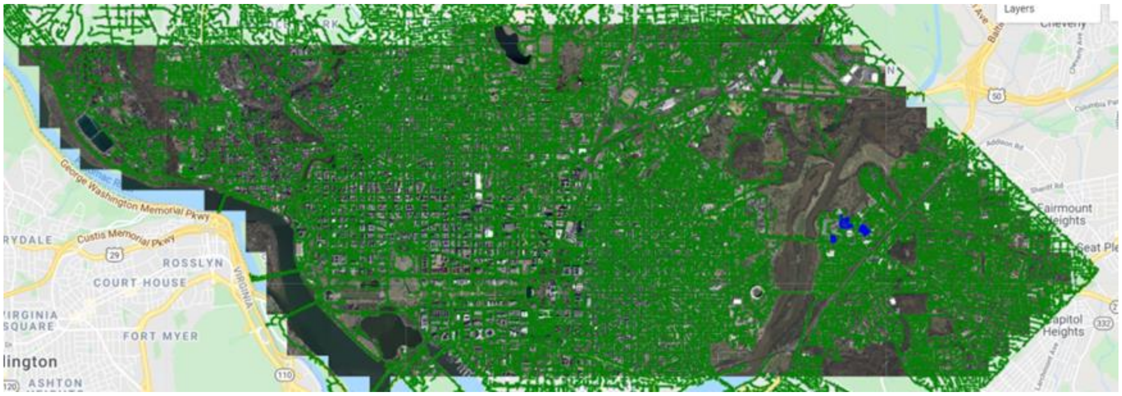


Figure 44. Wire map *truth* (Washington, D.C.)

2.4.2.3 Weight tuning and selection

Initially, four features (roads, railroads, waterbodies, and buildings) were selected as ones which provide *inference* to the graph network (Table 6). The edge weight values for the many-to-many Dijkstra's algorithm to use for building the grid are updated throughout the training. As a reminder, the default weight value is 1.0. Therefore, anything less than 1.0 promotes path growth, and anything greater than 1.0 discourages path growth.

Table 6: Weight tuning

Features	**Initial Weights**	**Run1**	**Run2**	**Run3**
Roads	0.10	0.75	0.85	1.0
Railroads	0.01	0.01	0.01	0.01
Waterbodies	20.0	20.0	20.0	20.0
Buildings	10.0	5.0	2.0	2.0

To determine the best set of features and weights, a parameter search was conducted. Since the set of parameters are small and the initial weights are assumed close to the final value, a grid search is feasible.

The results are also represented as a matrix of nodes whose values indicate the inclusion of the path, 1, or no path, 2. Calculating the *distance* or *error* between the two can be done using matrix math. We use the Frobenius norm to calculate the error, as shown in Equation 2.

$$\|A\|_F = \left(\sum_{i,j} abs\left(a_{i,j}\right)^2\right)^{\frac{1}{2}} \qquad 2$$

The norm is applied to the error matrix which is defined as the difference between the predicted and truth path matrices, as shown in Equation 3. In Equation 3, D is the error matrix, P is predicted grid matrix, and T is truth grid matrix. This directly correlates to how many predictions are incorrect, without regard for whether it is a false-positive or false-negative.

$$D = T - P \qquad 3$$

From this, the Matthew's Correlation Coefficient (MCC) can be computed to estimate the accuracy of the model.

2.4.2.4 Results on the Westchester County, NY data

In the first phase of the study, the cloud points generated were a set of predicted pole locations. In the second phase of the study, the set is a list of predicted wire locations. It is assumed that the same grid prediction model can be applied to this cloud of points. Each predicted wire location is then interpreted as a pole location by the algorithm. This assumption makes the grid prediction model more accurate as it follows the wire path more accurately.

The results of the grid prediction model applied on the Westchester predictions implemented in QGIS are shown in Figure 45 on the left side. On the right side, the model was applied to the actual pole locations. In both cases, the algorithm connected the dots to generate a prediction wire map.

Figure 45. Grid prediction result sample applied to predicted locations (left) and ground truth locations (right)

The generated shapefile of the prediction wire map has been imported on Google Earth Pro to show the results in 3D as depicted in Figure 46. All the wires were assumed to be at a height of 25m. The wire map is represented by the set of red walls.

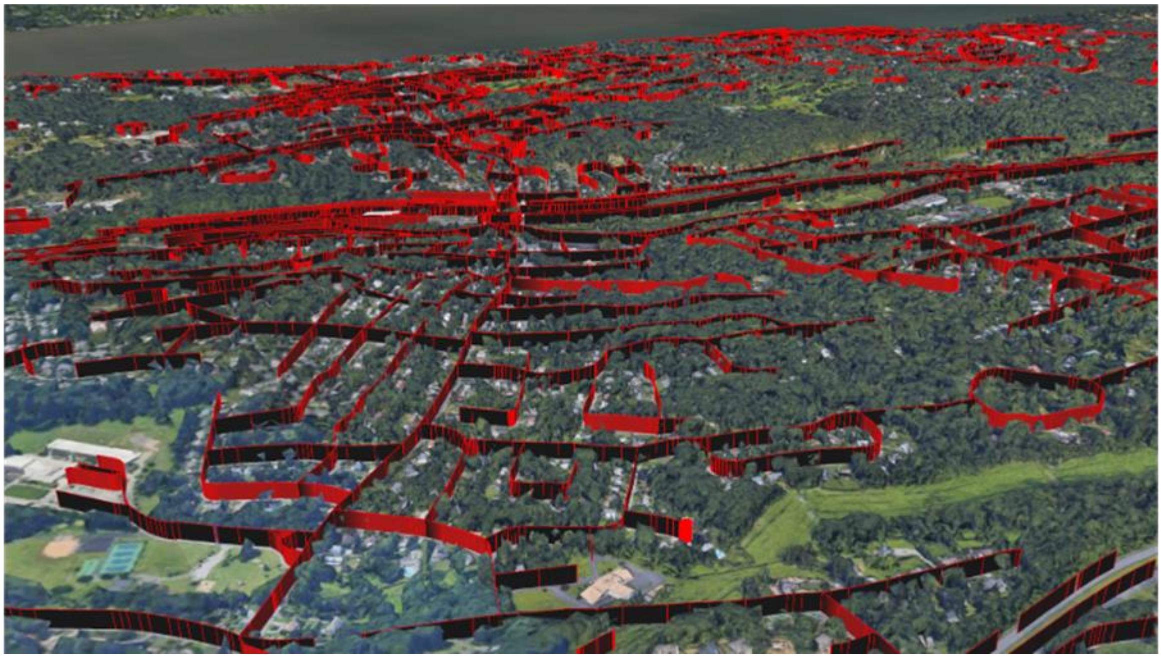


Figure 46. 3D visualization of a sample of the grid prediction result

To measure the accuracy of the global pipeline, both the predicted wire map and the generated true wire map were imported on QGIS. The true wire map was buffered by 10m using the data manipulation tools and subtracted to the predicted wire map. Depending on the result of the subtraction, the result highlights true and false negatives, and true and false positives. The result was plotted on QGIS, and each type of result was highlighted in a different color as shown in Figure 47. As each result is a set of lines, the length of each group gives an indication of the proportions. On the training set, there is 19738 km of false positives, 30734 km of false negatives, and 16533 km of true positives. Using this technique of measurement, it is difficult to evaluate the proportion of true negatives corresponding to the white background of the image. As such, the MCC cannot be evaluated using this method, a qualitative comparison is then performed.

Comparing the difference in proportion of results, the proportion of true positives can appear to be low. However, a closer look at Figure 47 shows that the false positives are close to the true positives and that most wires are detected. There are inaccuracies regarding the exact position of some wires that increase the number of false positive and negative predictions. Moreover, it appears that most of the true positives are wires located alongside roads. As explained in the previous sections, the street view imagery pipeline provides higher accuracy results. The UNET provides less accurate results due to the number of predictions generated. This does not represent the actual accuracy of the model as the results are bench-marked with a generated map based on real pole locations. Finally, to have an accurate measurement of the result, a true wire map would be necessary, yet in the Westchester County, such data does not exist.

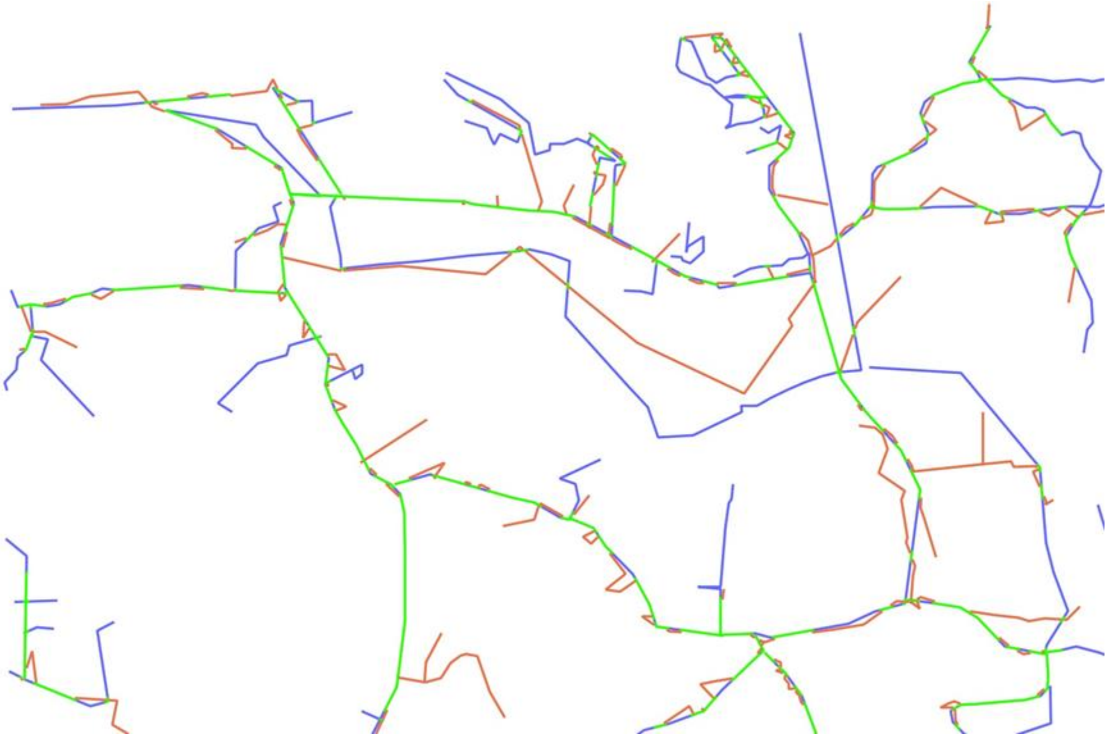


Figure 47. Confusion matrix for result of the pipeline for the second phase of the study (True positive: green, False negative: blue, False Positive: red, True Negative: white)

2.4.2.5 Final Results

The final results obtained are as follow:

- Prediction data and truth data results in a total 'distance' error of 328.84950965449224
- Truth data contains 44169 nodes, prediction data contains 84479 nodes
- This is a difference in 40310 nodes
- There were 74226 false positives, 33916 false negatives
- There were 10253 true positives, 1670325 true negatives
- Resulting in an MCC value of 0.13869536406096716

The wire map is shown in Figure 48 and displayed in 3D in Figure 49. Given the small training region for the predicted poles, those regions are promising and show that there is a lot of room for improvement that will be attempted in future research. The final MCC appears weak, but it is strongly correlated to the location of the predicted poles which had an MCC of 0.4 and the predicted pole can be anywhere in a 7-meter radius circle which then produces some uncertainty in the prediction.

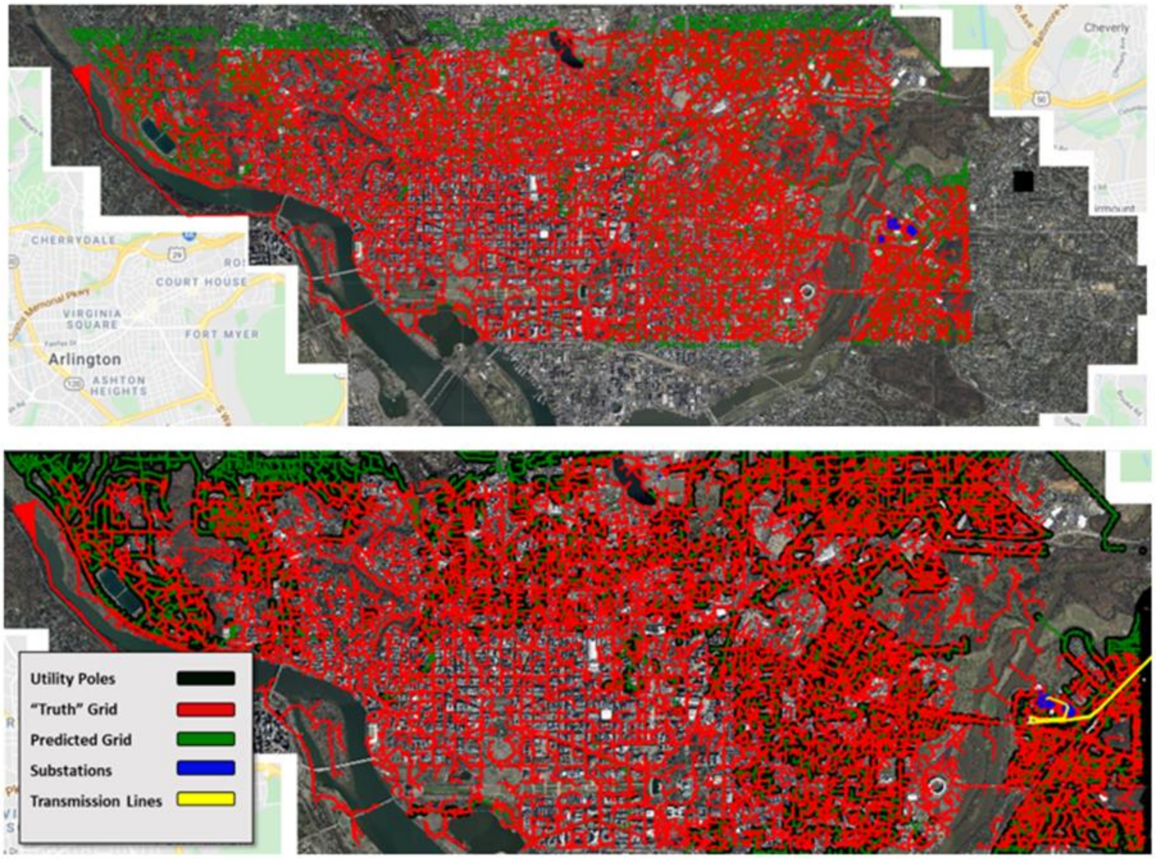


Figure 48. Final results for medium-voltage power lines in Washington, D.C. (first phase)

Figure 49. 3D Visualization of grid network in Google Earth (first phase of the study)

2.4.3 Confidence and sensitivity

It is expected that the capability of the network is related to the training dataset, parameters, and methods. For the pipeline and framework to be generalized and expanded to other regions, it is important that the pipeline includes the level of confidence for the predicted grid and that algorithms are robust enough for the datasets that are provided.

This is very important, particularly since the datasets will be very diverse for different regions. In particular, the grid prediction method is tested for the sensitivity to the features and weights used, as seen in Table 7. The total error for a subset of the test region is included in Table 7 as well. For the DC dataset, which is an urban and suburban environment, a majority of the grid network is located along the roads. Therefore, it is expected that the road feature weights are the most critical. As seen from the results, using a road feature weight below 1.0 is the primary method of improving the results. Any value below 1 promotes path growth, since the weights will lower to visit a neighbor node along that edge. For example, a road value of 0.10 promotes traveling along 10 road nodes rather than a single *feature-less* node. Additional weight tuning indicated that a higher weight for buildings prevented networks from cutting through structures along the network unless detections existed near the structure. Lastly, the increased weights on waterbodies led to a graph network that follows detections across bridges without cutting across the water elsewhere. This is not always the case in the real data, and the best parameters may need to be tuned for different locations and datasets. In the case that the features included here are not available for a specific region, it is assumed that the quality of the output will not be as high. Therefore, additional features may need to be detected from other predictive-based methods when the data is not available.

As explained previously, to extract the longitude and latitude of the wire poles from the prediction image, a filtering process was used, with a 7-meter grid, which is aligned with the path finding grid. The predicted pole can be anywhere in a grid cell, hence there is an uncertainty of seven meters for the location of the wire pole. Moreover, when training the network, a 7-meter radius circle was used to represent a pole. Similarly, the pole can be anywhere in this 7-meter radius circle. The worst-case scenario is if the predicted pole is located at the edge of the circle and the edge of a grid cell adding up to a 14-meter inaccuracy. Hence, assuming a perfect prediction, the total uncertainty of the presented pipeline is 14 meters for pole locations.

Table 7. Sensitivities to model weights

Features	**Case 1**	**Case 2**	**Case 3**	**Case 4**
Roads	0.10	0.75	0.85	1.0
Railroads	0.01	0.01	0.01	0.01
Waterbodies	20.0	20.0	20.0	20.0
Buildings	10.0	5.0	2.0	2.0
Error	272.01	271.26	271.52	273.60

2.5 Keys of improvement

The results presented here show great promise for the future. The techniques, framework, and specific code pipeline can successfully predict, within some level of uncertainty, the location of power line wires

2.5.1.1 Graph search improvement

Additional work has sought to utilize a parallel ripple-search (Brand, 2012) method to exploit multiple processors while searching. This method of improved convergence time may be necessary as the scale of accuracy increases and the number of graph nodes and edges increases. The time complexity of Dijkstra's Algorithm is $O(E + N \log N)$, with E as the number of edges and N as the number of nodes.

2.5.1.2 Additional models and training

In order to improve upon the current capability, future work can focus on including the implementation of one or more additional models for utility pole and wire detection. One area of interest is the use of object detection methods such as YOLO (Du, 2018) and Faster-R-CNN (Ren, 2015) for object detection and localization. This requires a bounding box in the dataset. The difficulty is that this would require the manual creation of a bounding box dataset and then manually drawing a bounding box around each pole.

Additional work should also consider issues with the implementation of other datasets such as off-nadir, different resolutions, and imbalanced dataset.

2.6 Summary

This work sought to provide rotorcraft pilots with information of wires in the surrounding area to prevent these often-fatal accidents. To do this, a detection and prediction approach was outlined, which first detects utility poles and then predicts the grid network in a region of interest. The pole detection step was tested with two methods: tile classification and semantic segmentation.

The grid prediction step was tested using many-to-many path planning. The outcome database contains a list of the location of wires in a shapefile. The full pipeline was tested in the Washington, D.C., Vermont, and Westchester County NY areas, showing the potential for a full wire database to be created using the proposed approach. The feasibility of extending this approach to other regions of the country would depend on the existence of satellite imagery database and street view imagery having a similar resolution to the ones used in this project. The next steps to fully implement the wire database are to integrate it on-board and to see how best to provide necessary information to pilots. The hope is that in the near future, the integrated database can prevent most wire strike accidents from occurring.

3 Identification of suitable EFBs for incorporation of wire database and development of wire proximity warning system

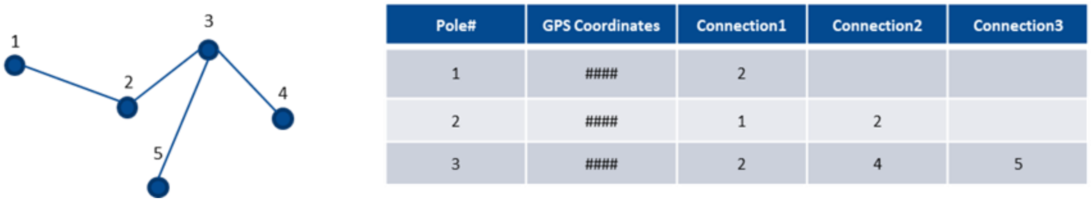
In Chandrasekaran et. al (Chandrasekaran R. , Payan, Collins, & Mavris, 2019), a set of potential electronic flight bag (EFB) manufacturers and their respective currently available systems were compared with the requirements for wire database management. The comparison table presented in Chandrasekaran et. al (Chandrasekaran R. , Payan, Collins, & Mavris, 2019) is displayed in Table 8. In this research, the selected EFB companies were contacted to establish a partnership.

Table 8. Electronic flight bags comparison

EFBs	**FlightDeck Pro**	**t.BagC2**[2]	**Aera 660**	**ForeFlight Mobile**	**FlyQ EFB 3.0**	**HeliEFB**	**RamcoEFB**
Features							
iOS	✓			✓	✓	✓	
Windows	✓	✓					
Other OS			✓				✓
Terrain Maps	✓	✓	✓	✓	✓		
Aeronautical Maps	✓	✓	✓	✓	✓	✓	✓
Dynamic Maps	✓	✓	✓	✓	✓		
Connection to Aircraft Avionics		✓					
Synthetic 3D Vision			✓	✓	✓		
Weather	✓	✓	✓	✓	✓		✓
Obstacle Avoidance	✓	✓	✓	✓	✓		
Augmented Reality					✓		
Split Screen					✓		
Flight Fog		✓			✓		✓
Weight & Balance Calculations						✓	✓
Risk Assessment						✓	✓
Maintenance Data							✓

The goal of this partnership was to implement the generated wire maps into their systems and then communicate the information to the rotorcraft pilots. A notion of the system requirements of the EFB is then required. More specifically, two main questions need to be answered: how to store the data, and how to display the data to the pilots.

For the first question of how to store the data, a certain structure for the database and a file format needs to be determined for the data to be compatible with their systems. An example can be that each wire pole coordinate is stored in a table and each row contains the list of the other wire poles they are connected to. This example is illustrated in Figure 50. In addition, a discussion on the limit of data file size needs to be conducted.



Pole#	GPS Coordinates	Connection1	Connection2	Connection3
1	####	2		
2	####	1	2	
3	####	2	4	5

Figure 50. Example of data structure for a wire map

Secondly, to answer the question of how to display the data, the actual visuals of the data, including color codes and shapes need to be decided as well. A possibility for the in-flight display would be to only show the data at altitudes where a wire strike is a risk. A notification protocol also needed to be established, such as should the risks be notified to the pilot only with visuals or should sounds be used.

Those questions must be discussed with an EFB company. A meeting with Foreflight, an EFB manufacturer, was done in the beginning of November 2020. However, the company did not maintain contact with our team. In the meantime, the team’s efforts to establish a contact within an EFB company were stalled due to the pandemic.

Instead, the team focused on producing a proof-of-concept visualization to demonstrate how the predicted wire map could be displayed on the pilot’s cockpit. Two concepts are presented: a 2D version and a 3D version.

3.1.1 2D EFB visualization

The first concept of visualization corresponds to adding the location of wires on the flight tracking display, the GPS. The location of wires would be displayed in the same way as roads are displayed for a car GPS. This visualization can be implemented directly with the results of the prediction pipeline. The outcome of the pipeline is a shape file containing lines indicating the presence of wires. In this proof-of-concept, the wires are displayed only when the vehicle is flying at a relatively low altitude where there is the highest risk for a wire strike to occur. Screenshots of the concept are shown from Figure 51 to Figure 53.

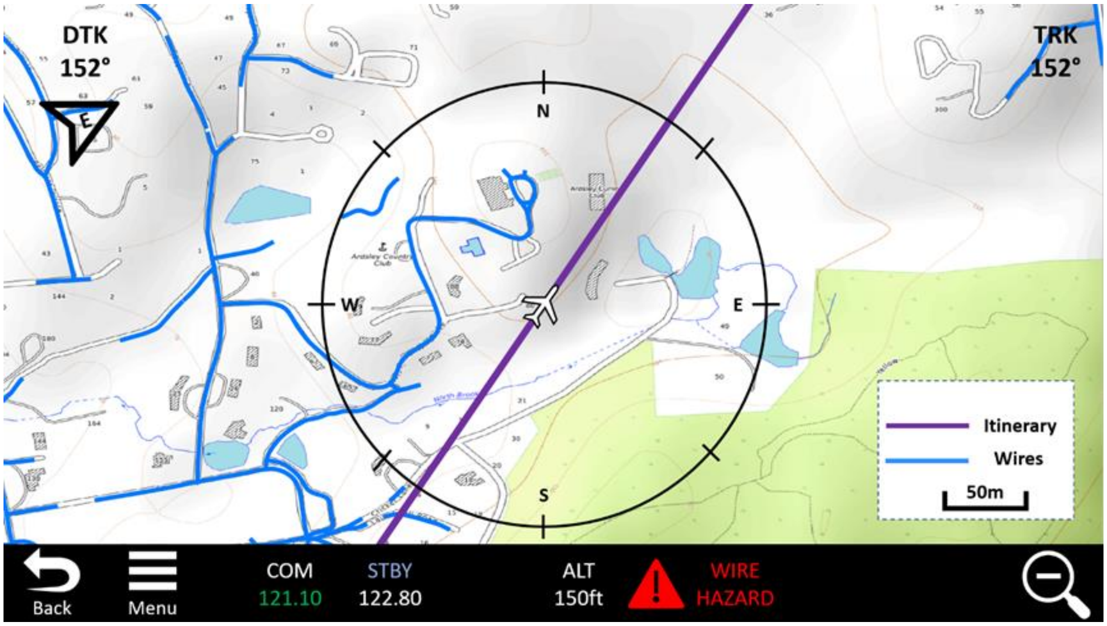


Figure 51. EFB 2D visualization concept - low altitude

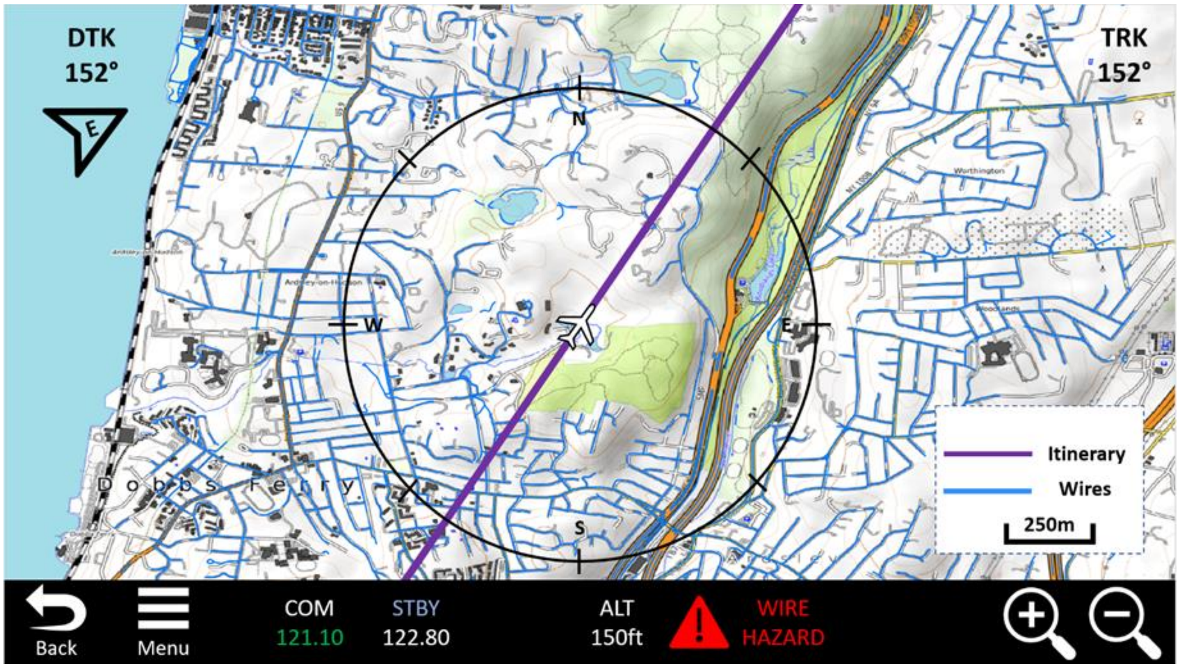


Figure 52. EFB 2D visualization concept - medium altitude

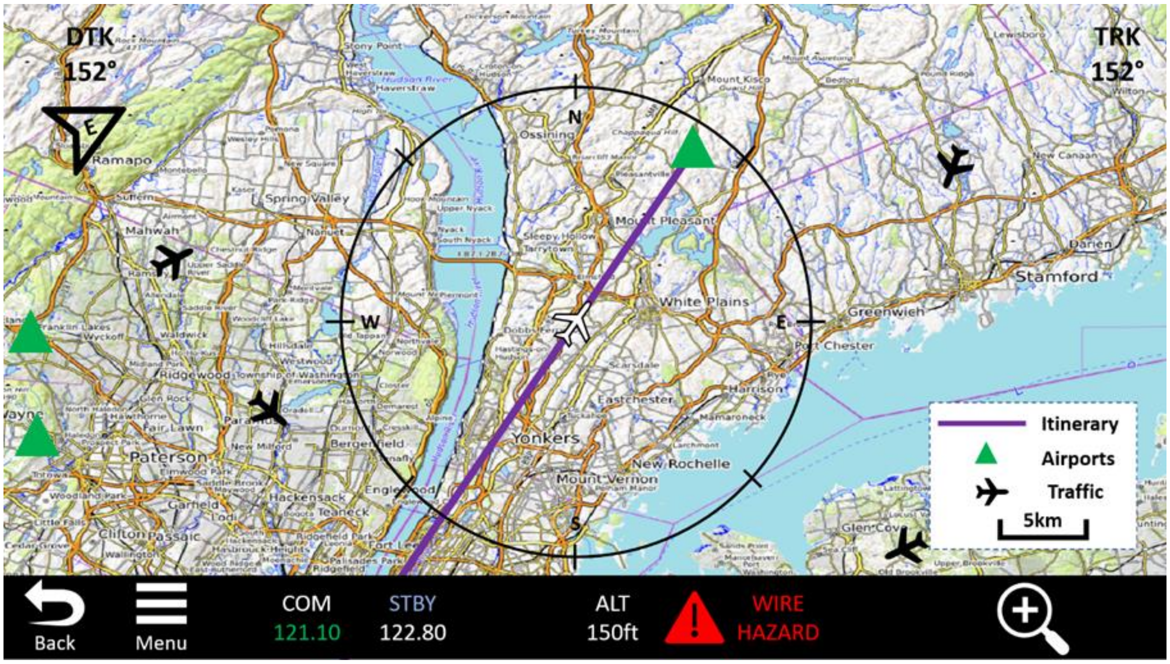


Figure 53. EFB 2D visualization concept - high altitude

In addition to this concept, a notification system needs to be established. A decision on how such notification would appear is needed. For example, when entering a zone with wires at a low altitude, a sound notification can ring, and a visual signal can appear on the dashboard. For future work, surveying rotorcraft pilots could help determine the best way to display the information.

3.1.2 3D EFB visualization

The second visualization concept presented is to include the location of wires in a 3D primary flight display (PFD). Such a view can help pilots to have a more precise idea of the relative location of the helicopter to wires. In addition to this, when maneuvering, additional visuals could appear to indicate a measurement of the distance and the location of the closest wire. However, the current prediction pipeline does not enable the possibility to generate a similar view. A prediction of the altitude of wires would be necessary. Obviously, the prediction results need to be very precise to not provide inaccurate information to the pilot. To make this view feasible it is best to combine the prediction pipeline with an on-board sensor fusion process to provide an accurate estimation of any wire or close obstacle location. A screenshot of this concept can be seen in Figure 54.

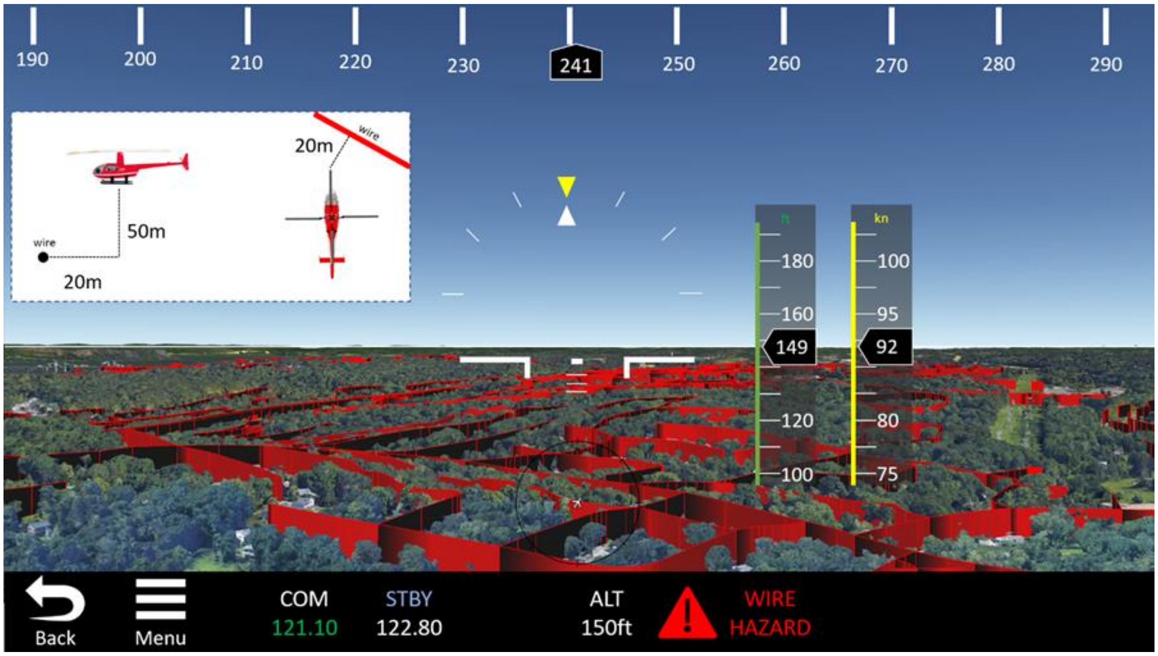


Figure 54. PFD 3D visualization concept

4 Physics-based modeling of wire cutter

In addition to the wire strike incident prevention technologies that are currently available or upcoming, methods to mitigate damage in the event of a wire strike incident are also available. The most widely used method today is the Wire Strike Protection System (WSPS) (Chandrasekaran R. , Payan, Collins, & Mavris, 2020) created by Magellan Aerospace, which consists of two components: a large scissor shaped cutter that is mounted to the top and bottom of the helicopter fuselage and a windshield deflector, as shown in Figure 55 (Chandrasekaran R. , Payan, Collins, & Mavris, 2020).

Figure 55. Wire strike protection system

In the event of a wire strike incident, the WSPS is designed to slice through the wire and decrease the impact to the helicopter and its controllability. According to Magellan Aerospace, the WSPS is most useful for impacts that happen in straight and level flight against horizontally strung wires (Magellan Aerospace, n.d.). In addition, the WSPS is available for a list of specified helicopters, the majority of which are heavy or medium weighted helicopters. One of the goals of this project was to investigate possible reasons that lightweight helicopters do not make use of passive protection systems as often as heavier rotorcraft. Some reasons could include cost, equipment weight, and a possible lack of effectiveness for a passive system on a lighter weight helicopter. Ultimately, the performance of the WSPS as a passive device heavily depends on the dynamics of the helicopter at the time of the incident. As a result, a lightweight or ultra-lightweight helicopter may not provide enough impacting force to allow the WSPS to successfully cut through the wire, and these helicopter types continue to be vulnerable in the event of a wire strike incident. In order to determine the usability of the existing passive systems on lighter rotorcraft, an analysis or simulation of a rotorcraft wire strike incident must be completed. The aim of a physics-based model of the WSPS is to quantify the requirements for successfully severing a wire in the event of a wire strike. With these requirements, it can be determined when the WSPS is effective, and what the limits of the helicopter weight and state variables are for a wire strike event.

4.1 Current technology and goals

Before exploring the wire strike modeling side, some previous research was completed regarding prior studies on the current capabilities of the WSPS. These studies gave insight into what has already been explored in this area of research, as well as some results to compare with the results of the final wire strike model of this study. According to Magellan Aerospace’s website (Magellan Aerospace, n.d.), the WSPS has the ability to reduce: risk in forward, level flight, , the chance that the wire penetrates the windshield when impacting horizontally strung wires, the possibility that the wire contacts the main rotor, and the possibility that wires become ensnared in the landing gear. These risk reductions correspond to the efficacy of the windshield deflector, upper cutter, and lower cutter, respectively. Those three components make up the full Wire Strike Protection System. In addition, the WSPS is rated to destroy three-eighth inch diameter steel cable wires at a maximum tensile strength of twelve thousand pounds (Magellan Aerospace, n.d.). As a method of verification, Magellan Aerospace cites multiple tests completed by the U.S. Army at NASA, where they complete experiments using a drop test facility for helicopters. A few of the results of those experiments are listed in the section below, along with a discussion of how these experiments give insights to the ability of the WSPS in the tested situations.

4.1.1 Prior research into WSPS efficacy

Experimental testing of the WSPS as installed on the OH-58A was completed in the 1980s (Burrows L. T., 1980). The OH-58A Kiowa is manufactured by Bell, and it has a loaded weight of three-thousand pounds, making it a lightweight helicopter. The results of this experimental test are particularly relevant to this project with regards to determining how wire cutters perform on lighter weight helicopters. To simulate the effects of the wire strike without a piloted test, the Applied Technology Laboratory (ATL) made use of a helicopter drop tower which swung the helicopter on a cable towards a wire mounted at the bottom of the helicopter’s trajectory as shown in Figure 56 taken from Burrows (Burrows L. T., 1980).

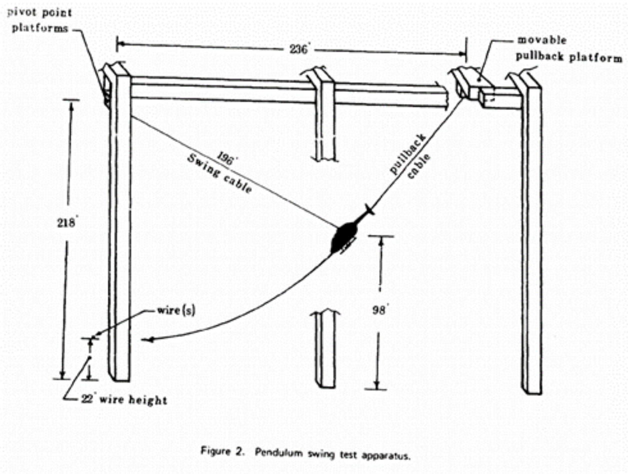


Figure 56. Helicopter drop tower experimental setup

The helicopter was retracted a set distance that estimated its velocity upon impacting the wire. The wire was positioned at various angles to experimentally simulate the helicopter impacting the wire from different angles. Ultimately, the results of the study were impacted by the fact that the wire was more likely to hit the windshield protector in this experimental configuration than the wire cutter itself. In this case, the results of the study provided less information about the capabilities of the upper and lower wire cutters within the full WSPS. In addition to these tests, experiments were also completed with the upper wire cutter attached to the fuselage of an OH-58A helicopter that was mounted to a truck. The truck was driven into wires at various speeds and angles, and the result was that the WSPS was effective.

Another similarly executed experiment was completed for the AH-1S Cobra, due to the fact that this rotorcraft model was much different from the OH-58A tests that were initially completed in Burrows (Burrows L. T., 1982). The AH-1S has more equipment, and therefore more potential for wire snags. The results of the test were similar in the fact that the WSPS was able to ease the process of cutting or snapping the wire during a wire strike incident. However, due to the equipment and weapons attached to the AH-1S, it was determined that the WSPS would be less effective for protecting against wire strikes in various locations on the rotorcraft. Although there is extensive work on the experimental side of verifying the WSPS, it is not evident if a physics-based analysis or simulation for the wire strike incident has been developed. Because this physical interaction between the WSPS and the wire is not well defined thus far, a physics-based approach to defining the efficacy of a wire cutter would improve the understanding of the limits of the WSPS. In addition, an extensive wire cutting model would also aid in design and verification of new WSPS designs.

4.1.2 Reports of WSPS damage mitigation

An example of the WSPS preventing a major helicopter accident has been detailed in an accident report posted by Aerossurance (Aerossurance, 2018). The helicopter described was a firefighting helicopter taking part in a mission when it struck a wire. There was a Wire Strike Protection System mounted on the helicopter which was able to make impact with the wire as shown in Figure 57 taken from Aerossurance (Aerossurance, 2018).

Figure 57. Wire cutter after a wire strike incident

Figure 57 shows the remnants of the wire caught by the WSPS . After the incident, the pilot was able to land and access the damage. This incident shows that the WSPS can be a valuable tool to reducing the severity of wire strike incidents. The helicopter involved in the incident described is a lightweight helicopter created by Airbus with a wire cutter attachment made specifically for that helicopter model.

In addition to this description of how the WSPS was able to protect the lightweight Airbus helicopter from a strike, there are a few instances of wire strikes involving the WSPS as listed by the NTSB. While not many of these accidents explicitly state that there was a wire strike with a WSPS installed on the helicopter, there are a few reports that do provide more information about the applicability of the WSPS in certain accidents. The specifics of these reports are discussed further in the section on validation of the wire strike simulation, where the information provided is considered as a verified scenario for the effectiveness or ineffectiveness of the WSPS.

4.2 Physical properties of a wire strike impact

In order to develop an approach to modeling the physics of the interaction between the wire, the WSPS, and the helicopter, the wire strike is first decomposed into a sequence of three physical events as shown in Figure 58.

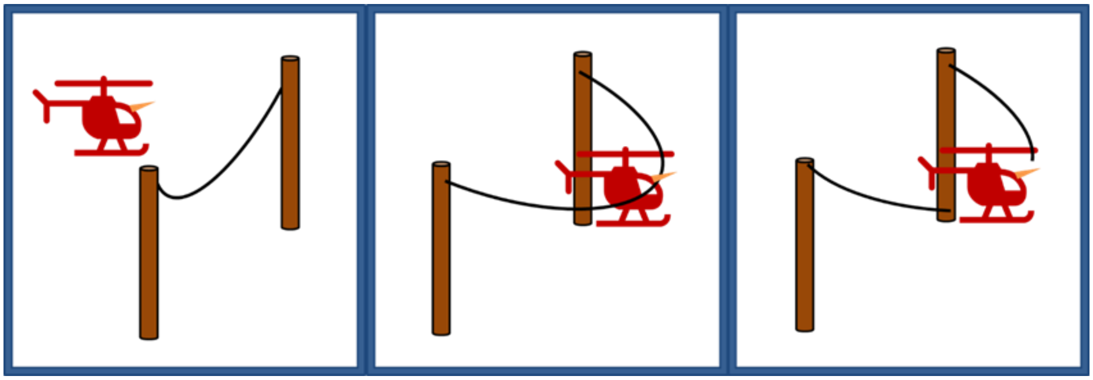


Figure 58. Visual of the wire strike process

The wire strike starts when the helicopter is just about to impact a wire, represented by the first event depicted on the left side of Figure 58. In this moment before the impact of the wire and the helicopter WSPS system, the helicopter has some initial velocity and orientation. These values, along with the helicopter mass, will be important for determining how much force is put into cutting the wire upon impact. To simplify the various possibilities, the assumption is made that the wire impacts the WSPS directly and not somewhere else on the helicopter. This can be expanded further to consider the use of the windshield deflector that can also aid in severing the wire. In addition, the assumption is made that there is no pilot control input during the wire strike incident, but this assumption can be expanded on as well with a more encompassing helicopter dynamics model. Once the helicopter and WSPS begin to impact the wire as shown in the second event depicted in the middle of Figure 58, there are a variety of phenomena happening to the wire. One is that the helicopter is applying force to the wire and in turn the helicopter is slowing down due to the wire reaction force. Essentially, the helicopter is imparting some of its energy to the wire, and some energy is dissipated in the form of heat, noise, friction, and other losses. In addition to this dynamic impact, the blades of the WSPS are applying a shear force to the wire, possibly severing it. In addition to the shear force, the impact is also increasing the tension in the wire, possible exceeding the wire tensile strength, which is decreasing as the wire is severed. Finally, the rotorcraft is either able to sever the wire and continue on as visualized in the third event depicted on the right side of Figure 58, or the wire is not severed, inhibiting the helicopter movement and possibly leading to a serious incident or an accident.

Based on the decomposition from Figure 58, there are some main physical disciplines that can be investigated when trying to remake the incident in a simulation. The first interaction considered when developing the structure of the final simulation was that of the wire and the wire cutter portion of the WSPS. The goal of the final simulation was to determine if the wire would be successfully severed by the WSPS, and the contact between the wire cutter and the wire would provide the requirements for the wire to ultimately be cut. The wire and the wire cutter must contact each other with some amount of force for enough shear force to be generated to completely sever the wire. To determine how much force is required, the next discipline that needs to be considered is that of the wire dynamics. When a wire strung with tension, as powerlines are, is impacted, how much energy is dissipated by the wire, and ultimately, the resulting force on the wire, are considerations that are needed to determine if the wire is cut. In addition, another quantity of interest is the tension induced in the wire by the impacting forces, as a snapped wire would also complete the purpose of the WSPS.

The final discipline of interest is the behavior of the helicopter after the impact. If the helicopter which struck the wire is not controllable, despite severing the wire, then a crash due to the wire strike incident may still occur. Capturing this aspect of the incident as well is important to determine the safety of the passengers overall after an incident. With these topics in mind, another notable requirement for the simulation is that it should be used to make quick decisions on wire cutter designs. Ultimately, some more detailed considerations for the physical properties of the contact between the surface of the wire cutter blades and the wire itself were omitted to meet this requirement. Given this requirement and the disciplines involved in the wire strike as a whole, the following section details the methodologies used to model each portion of the overall wire strike simulation.

4.3 Survey of research and literature

4.3.1 Mechanics of cutting

The initial goal was to focus on the interaction between the blade of the wire cutter and the wire by modelling the contact mechanics and crack growth as the helicopter impacted the wire. Ultimately, the focus shifted to creating a more encompassing model considering the various components of the energy balance throughout a wire strike incident. As described previously, a major portion of the wire strike incident is the interaction between the wire cutter and the wire, as this contact will ultimately determine if the wire is severed. The physics of this interaction was explored by first looking into theoretical and experimental work in wire cutting in various disciplines. Looking more closely at the contact between the wire cutter and the wire upon impact, it can be determined that the shear force imparted on the wire by the wire cutter is ultimately the important force to consider regarding whether the wire cutter will cut through the wire, as demonstrated by the shear cutting process of sheet metal in Figure 59 (CustiomPartNet, n.d.). To better understand this shear stress, research was completed regarding cutting materials in different disciplines. Regarding the mechanics of cutting, there were two main disciplines that have been researched extensively: orthogonal cutting of metal and shearing through soft materials. The orthogonal cutting of metal and chip formation relates mostly to the machining of metals, and shearing through soft materials is applicable to surgical applications including surgical scissors. While each of these types of studies have interesting information regarding the mechanics of cutting and fracture mechanics, the application to a wire strike incident is not immediately apparent. A discussion of these cutting applications and the applicable information from them as well as the results of more applicable studies into cutting mechanics are discussed in this section.

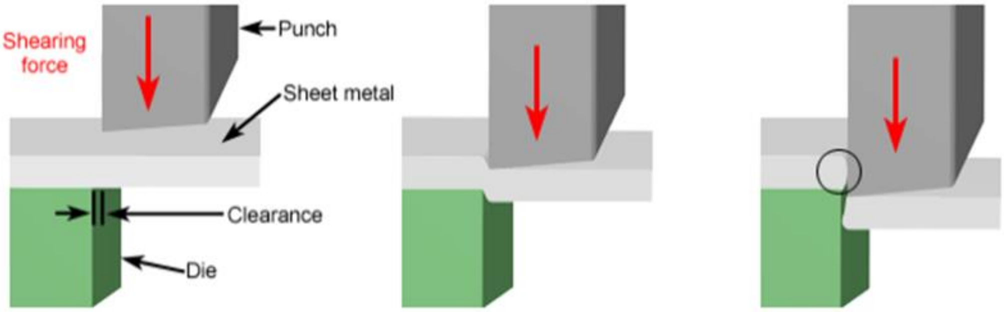


Figure 59. Shear cutting in sheet metal

The study on “Fundamentals of Cutting” by Williams and Patel in 2016 (Williams, 2016) discusses the energy required to fracture and cut a layer of a material. They focus on chip formation, and they assume a low velocity such that kinetic energy was not considered which decreases the applicability to the wire strike case, as the wire cutter is impacting the wire at the speed of the helicopter. The researchers determined that the energy balance for the cutting process was described by Equation 4.

$$dU_{ext} = dU_c + dU_f + dU_d \tag{4}$$

In Equation 4, U_{ext} represents the external work done, dU_c represents the fraction energy, dU_f represents the friction energy, and dU_d represents the plastic dissipation energy (Williams, 2016). When the force is large enough, then the cutting process continues as shown in Figure 60 (Williams, 2016), and the trimmed layer can also break into chips. In an attempt to apply this energy model to the wire cutter configuration, some of the parameters in the diagram were considered for the wire cutting case.

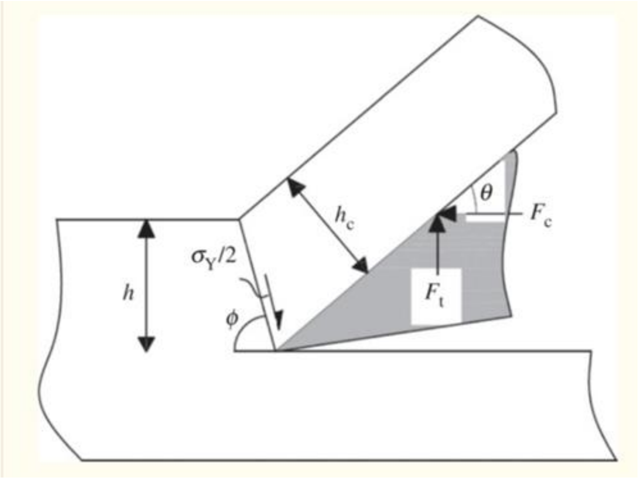


Figure 60. Chip formation in cutting a thin layer of a material

The parameter h would correspond to half of the length of the wire, and the chip formation aspect would not have much of an effect on cutting through the full diameter of the wire. While this paper provided some information regarding the energy required to shear through a material, it is not immediately apparent whether it can be applied to the wire cutting case. As a result, this model was not followed as the final consideration for cutting mechanics.

Another research paper called “Modeling the Force of Cutting with Scissors,” written by researchers Mahvash et al. discusses both the fracture mechanics of cutting as well as the blade sharpness as aspects of cutting mechanics (Mahvash, et al., 2008). Mahvash et al. model the torque and crack tip position related to the process of cutting a material with scissors. They consider an energy model, where the conservation of energy is as in Equation 5.

$$dW_e = dW_A + dU \tag{5}$$

In Equation 5, dW_e is the external force which is calculated using the torque and opening angle of the scissors, dW_A is the work of fracture which is related to the fracture toughness of the material and the area separated assuming a very sharp blade, and dU is the change in elastic potential (Mahvash, et al., 2008). This model was used to predict the required force applied by the user of the scissors to cut a material and was tested experimentally as well. Ultimately, this model provides another application for an energy model including fracture mechanics, however it is also not fully applicable to the wire cutting case. The main differences between this model and the wire strike case are the fact that kinetic energy involved in the impact is not considered and the model focuses on the application of torque from the scissors, whereas the blades of the WSPS are not rotating. See Figure 61 (Mahvash, et al., 2008).

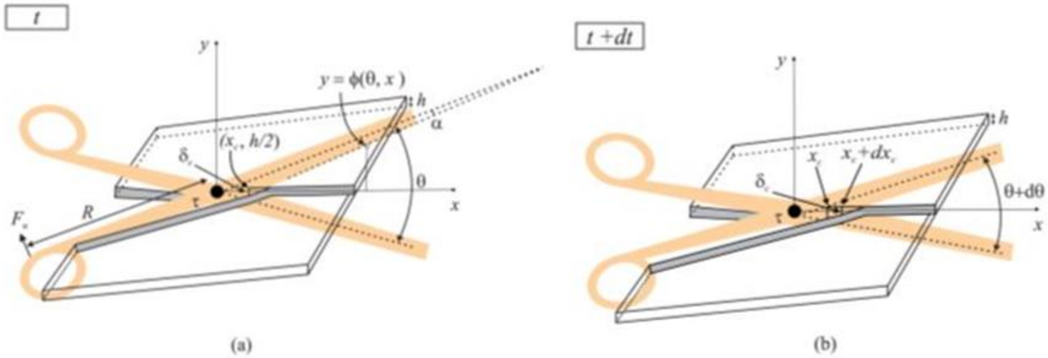


Figure 61. Shearing with scissors

Despite this literature search into the involvement of fracture mechanics and energy methods on determining the requirements for cutting through a given material, there has not yet been a research application that is fully applicable to the wire strike case. “Mechanics of a shear cutting process” by Meissner models and experimentally tests the forces involved in shearing through a given object (Meissner, 1997). The experimental setup in this study involves an angled blade propelled by a load cell, and a second straight, stationary blade beneath as shown in Figure 62 (Meissner, 1997). A material is placed between the two blades, and the upper blade is propelled into the lower blade.

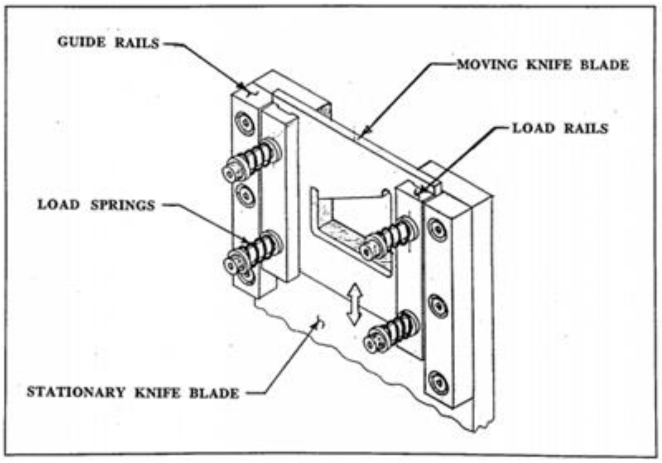


Figure 62. Experimental assembly for shear cutting mechanics

Meissner creates an analytical model of this experiment, as well as completes the experiment for various loading cells. The force required to cut through a material is calculated using the formula for shear stress shown in Equation 6.

$$F = \tau_{ult} * A \tag{6}$$

In Equation 6, τ_{ult} is the ultimate shear strength of the material being cut, and A is the shear area as defined in Figure 63 (Meissner, 1997). The shear area is defined in this research by the formula $h^2/[2 * tan\phi]$, given the subject material as a polycarbonate sheet with a rectangular cross section (Meissner, 1997). After estimating the force required to cut the various polycarbonate samples, experimental testing was completed that determined what forces were used to cut through each sample of material.

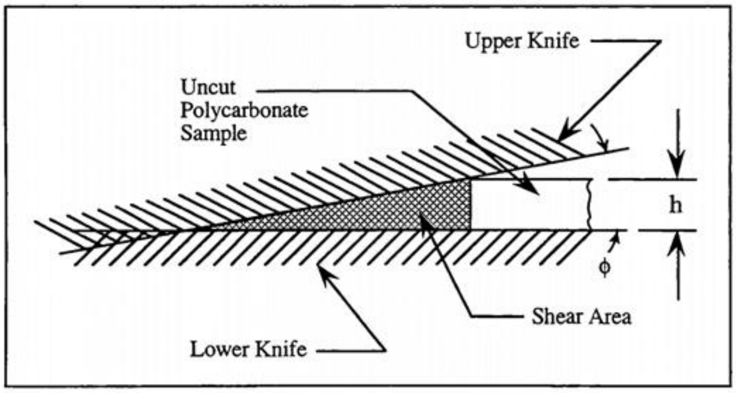


Figure 63. Definition of the shear area

As with the previous models, this model was also considered for its applicability to the wire strike case, and contrary to the previous models, this case appears to be much more relevant to the problem at hand. While the other models assumed that kinetic energy was not a factor in the cutting process, in this study, the force behind the moving upper knife is a key component to determining if the material is cut. This aligns closely with the dynamic nature of the wire cutter and wire interaction during a wire strike, as the force behind the wire cutter is the main actor in severing the wire with a passive protection system. In addition, the experimental setup involves two blades with an angle between them, similar to how the passive WSPS is set up. While the upper blade of the WSPS does not push down on the wire as the upper knife does in this example, some similarities can be drawn between the two situations. As the wire approaches the intersection between the two blades of the WSPS, a similar picture to Figure 63 can be seen in Figure 64, where the shear area is the area between the wire and the intersection of the two blades. As the WSPS cuts into the wire, this area decreases further, just as it would in the experimental setup of this study. In the context of determining whether a wire will be severed given the parameters of the impacting scenario, the variable to solve for would be the estimated force required to cut the wire assuming that the geometry of the WSPS and the properties of the impacted wire are provided. Given these parallels and the fact that this model is easily implementable into the final wire strike simulation, it was chosen to represent the cutting mechanics aspect of the WSPS.

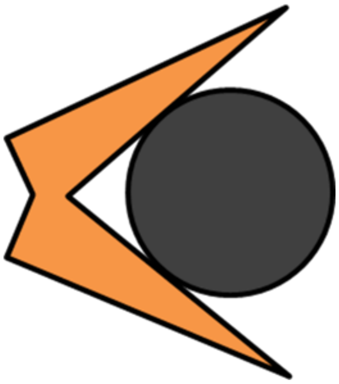
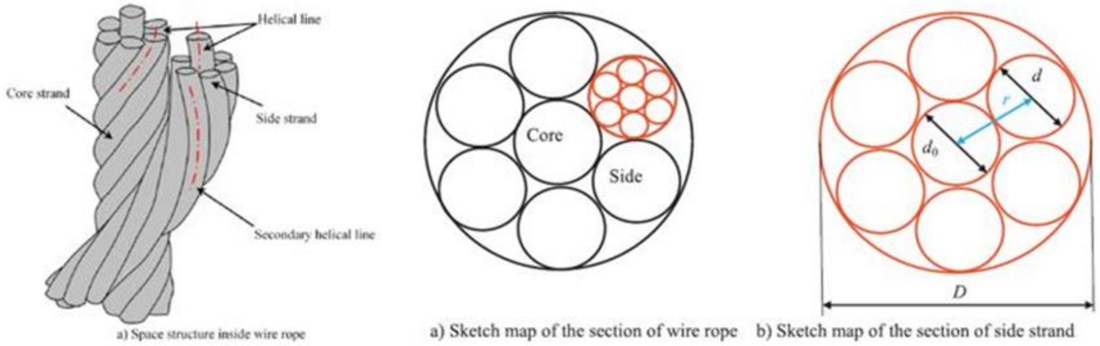


Figure 64. Representation of the blades of the WSPS around a wire

4.3.2 Wire impact dynamics and wire mechanics

Now that a determination for how much force is needed to sever through the wire has been chosen, the amount of force generated by the impact between the helicopter and the wire needs to be calculated and compared to this required force. This interaction force will be determined by considering the dynamics of the wire as it is being impacted. There are a number of different types of cables that rotorcraft may impact, ranging from power lines at very low altitudes to large cables at higher altitudes. The type of cable being impacted is another input of the model, however the initial stage of the model is focusing on wire ropes. Wire ropes are structurally and dynamically complex. They consist of a system of thinner wires that are twisted around each other, and these resulting strands of twisted wires can then be twisted together recursively in multiple layers to result in the final wire rope. Therefore, information regarding the mechanical properties of a wire rope were important to use in calculations of cutting mechanics and wire dynamics that will also be discussed. The paper “Mechanics model and its equation of wire rope based on elastic thin rod theory” by Wu and Cao, describes the mechanical properties of the wire based on its geometry (Wu & Cao, 2016). A visualization for the geometry of a wire rope consisting of multiple side strands wrapping around a core strand to create new core and side strands is shown in Figure 65 (Wu & Cao, 2016).



a) Space structure inside wire rope a) Sketch map of the section of wire rope b) Sketch map of the section of side strand

Figure 65. Visualization of the structure of a wire rope

Based on the geometry of the specified wire, such as the number of side strands in each strand layer of the wire and the diameter of the different types of strands, researchers in this paper made estimations for the values of the elastic modulus of the rope as a whole, by beginning with the elastic modulus of the smallest strand of core and side strands, and combining these layer by layer into the final elastic modulus of the wire. The elastic modulus of a side strand is given by Equation 7.

$$E_s \approx (d_0/D)^2 E_0 + \sum_{i=1}^{n} m_i (d_i/D)^2\, E_i sin\alpha_i \tag{7}$$

In Equation 7, d_0 is the diameter of the side core wire, D is the diameter of the strand, E_0 is the elastic modulus of the core strand, d_i and E_i are the diameter and elastic modulus of side strand i respectively, α_i is the helix angle of the side strand, and m_i is the number of side strand wires in that layer. Once the elastic modulus of the side strand is determined, the formula can be used recursively for each layer of the wire starting with the strands being made of solid material, and Equation 8 is used to calculate the elastic modulus of the wire as a whole.

$$E_r \approx (d_0'/D')^2 E_{s0}' + \sum_{i=1}^{n} m_i' (d_i'/D')^2\, E_{si}' sin\alpha_i' \tag{8}$$

The mechanical behavior of the wire under loading is modeled in the book *Cable Structures* by Max Irvine. It details how the tension in a guy wire under a static load, or a cable that is connected from the ground to the top of a pole, is a function of both the wire properties and the displacement of the wire under loading (Irvine, 1981). A visual of the setup can be seen in Figure 66 (Irvine, 1981), in which there is a point load applied to a wire that is being supported on each end.

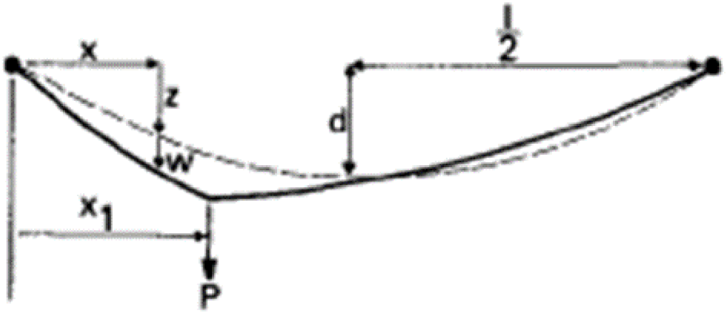


Figure 66. Wire supported at each end and influenced by a point load

Using the point load, its location along the wire, and the displacement of the wire, the increase in horizontal tension can be determined. In the context of an energy model, the elongation of the wire is an important factor for determining the elastic energy in the wire as a load is applied to it, and with regards to the mechanics of the wire after impact, the increase in tension can determine if the wire will snap through the tensile strength being exceeded. The relationship between the point load, the displacement, and the tension is given by Equation 9.

$$(H+h)\frac{d}{dx}(z+w) = P\left(1+\frac{x_1}{l}\right)+\frac{mgl}{2}\left(1-\frac{2x}{l}\right) \qquad 9$$

In Equation 9, H is the initial tension in the wire, h is the increase in tension, z and w are the displacements of the wire as defined in Figure 66, P is the point load, x_1 is the distance to the point load from the support, and l is the distance between the two wire supports. Using this model, the increased tension in the wire as a result of the force of the impact can be estimated (Irvine, 1981).

When it comes to a dynamic load, Xu et al. developed a physics-based model to determine the reaction force over time from a wire rope upon impact with a high-speed object as seen in the setup shown in Figure 67 (Xu, Dong, Zhang, & Zhang, 2014). Xu et. al modeled the wire being impacted as a multi-body of fundamental cylindrical elements (FCEs) shown in Figure 68.

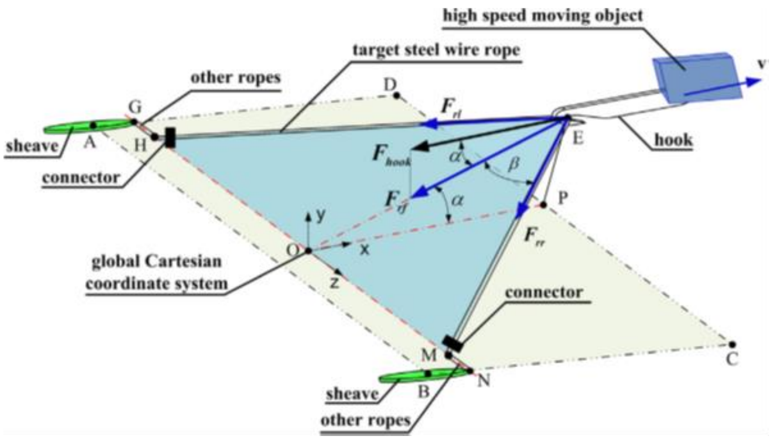


Figure 67. Experimental setup for wire dynamics model

Each small element in the rope has its own dynamics, and the generalized force, position, and velocity of each element is connected through a spatial constraint, as shown in Equation 10.

$$P_i = -K(q_i - q_{i+1}) - C(\dot{q}_i - \dot{q}_{i+1}) + P_{presi} \tag{10}$$

In Equation 10, P_i is the generalized force on the i_{th} FCE, K and C are the stiffness and dampening matrices respectively, q_i is the position of the i_{th} FCE, and P_{presi} is the preset force on the i_{th} FCE.

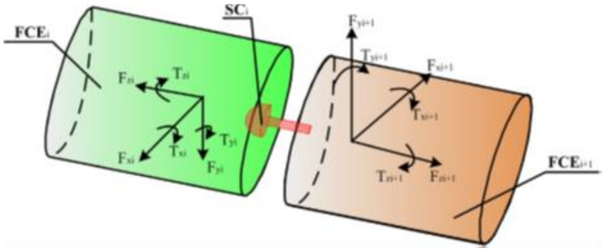


Figure 68. Fundamental cylindrical elements (FCEs)

The generalized forces on the FCEs in contact with the impacting object are determined by Equation 11 which uses the velocity and position of the object.

$$P_{\{Fxm\}} = \begin{cases} 0, q_x \geq w_{x0} \\ K_m(q_{x0} - q_x)^{e_m} - C_m \frac{dq_x}{dt}\left(\frac{q_x - q_{x0} + d_m}{d_m}\right)^2 \left(3 - \frac{2(q_x - q_{x0} + d_m)}{d_m}\right), q_{x0} - d_m < q_x < q_{x0} \\ K_m(q_{x0} - q_x)^{e_m} - C_m \frac{dq_x}{dt}, q_x \leq q_{x0} - d_m \end{cases} \tag{11}$$

In Equation 11, P_{Fxm} is the impacting force in the x direction, K_m is the contact stiffness coefficient, C_m is the contact dampening coefficient, q_{x0} and q_x are the initial and instantaneous distances between the impacting object and the rope respectively, and d_m and e_m are constants related to the contact between the rope and the impacting object. The y and z components of the impacting force are calculated similarly. The contact modeling is shown in Figure 69 (Xu, Dong, Zhang, & Zhang, 2014).

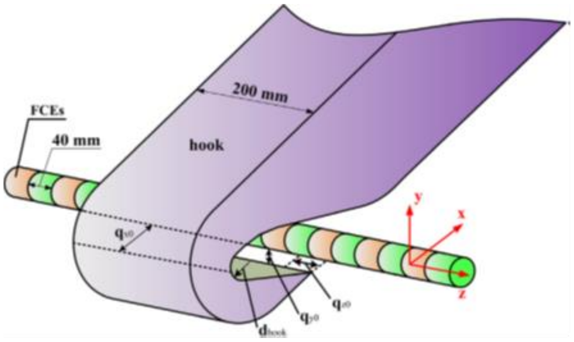


Figure 69. Contact area between the hook and the wire rope

The generalized force on the remaining FCEs in the rope are derived from the spatial constraint equation and the Lagrange equations. Lastly, the final reaction forces of the rope onto the impacting object are calculated based on the generalized forces in the FCEs, as well as the effects of the kink-wave in the rope that is generated upon impact. Researchers in this study also consider the fact that upon a dynamic impact, a wave is induced in the rope that affects the resulting force back on the impacting object. The angle β is used to apply the effects of the kink-wave to the resultant force, and it is calculated based on the wave of stress propagated throughout the rope, as shown in Equation 12.

$$\beta = \begin{cases} atan\left(\frac{\int_0^t wdt}{\sqrt{\left(\int_0^t v_m dt\right)^2 + L_{EP}^2}}\right), 0 < t < t_w \\ atan\left(\frac{L_{ON}}{\sqrt{\left(\int_0^t v_m dt\right)^2 + L_{EP}^2}}\right), t \geq t_w \end{cases} \qquad 12$$

In Equation 12, w is the velocity of the kink-wave shown in Figure 70 (Xu, Dong, Zhang, & Zhang, 2014), v_m is the velocity of the object, L_{EP} and L_{ON} are distances as defined in Figure 67, and t_w is the time that the kink wave propagates to the sheave. The impacting forces and this angle are used to calculate the resultant force from the wire back onto the impacting object, as explained in Equation 13.

$$F_{rf} = F_{rl} cos\beta + F_{rr} cos\beta \qquad 13$$

In Equation 13, the resultant force is the combination of the left and right rope forces as shown in Figure 67. This resulting force from the rope back onto the impacting object is the force that will be compared to the force required to cut the wire. The same methods as the ones used by the authors in this study are being implemented in the wire strike model and simulation (M&S) environment to find the reaction forces during impact.

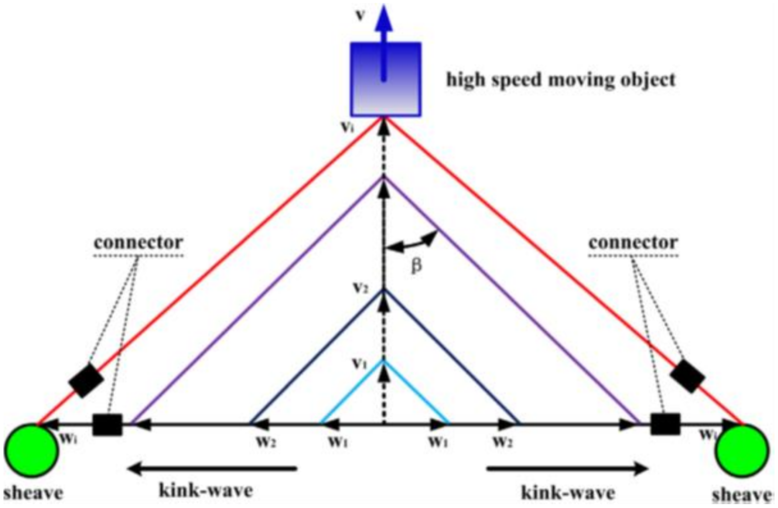


Figure 70. Visualization of the kink-wave propagation

4.3.3 Helicopter dynamics

In order to adapt the previously shown wire dynamics model depicted in Figure 67 to the helicopter wire strike case, the dynamics of the impacting object will need to be replaced with the dynamics of the helicopter. In the equations for impacting force, the velocity of the impacting object should be dependent on the state of the helicopter upon impact. In addition, once the reaction forces of the wire are calculated, they can be used as inputs to determine the new state of the helicopter given the reaction force from the wire. As opposed to a moving block which has relatively simple dynamics and no considerations for other forces, a helicopter must also take into account the forces from the rotors and stabilizing forces built into its design. For the purpose of including these other forces into the wire strike model, an applicable rotorcraft simulation was considered to interact with the wire dynamics. Rotorcraft simulation is a generally well explored discipline, and a helicopter dynamics model developed by NASA (Chen, 1979) and shown in Figure 71 is implemented within the wire strike M&S environment.

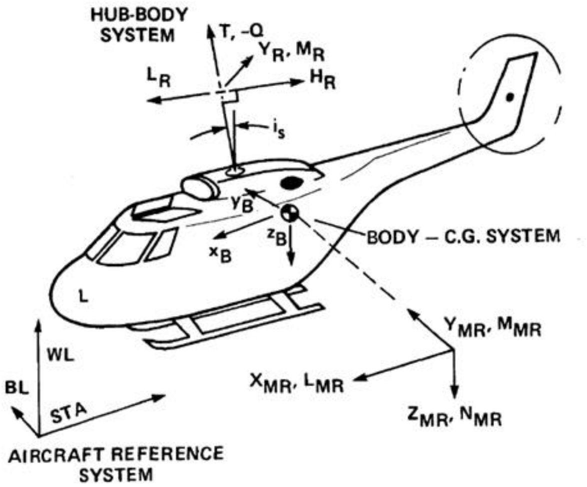


Figure 71. Rotorcraft reference axes

This model considers the dynamics of the main rotor flapping as well, making it a nine degree of freedom dynamics model. In terms of forces from the helicopter, the model includes forces from the main rotor of the helicopter, the tail rotor of the helicopter, the horizontal stabilizer and vertical fin, and the fuselage aerodynamics. The model also includes an RPM governor and the cockpit controls and cyclic control phasing. Each of these force considerations are determined for the helicopter using the helicopter state, control parameters, and a large list of helicopter parameters for the mass properties of the helicopter, and properties of each of its major components. The forces from each component are all translated to the axis at the center of gravity of the helicopter along with the external forces and moment of force, and these are used to solve for the helicopter state using a six degree-of-freedom rigid body equations as exemplified in Figure 72.

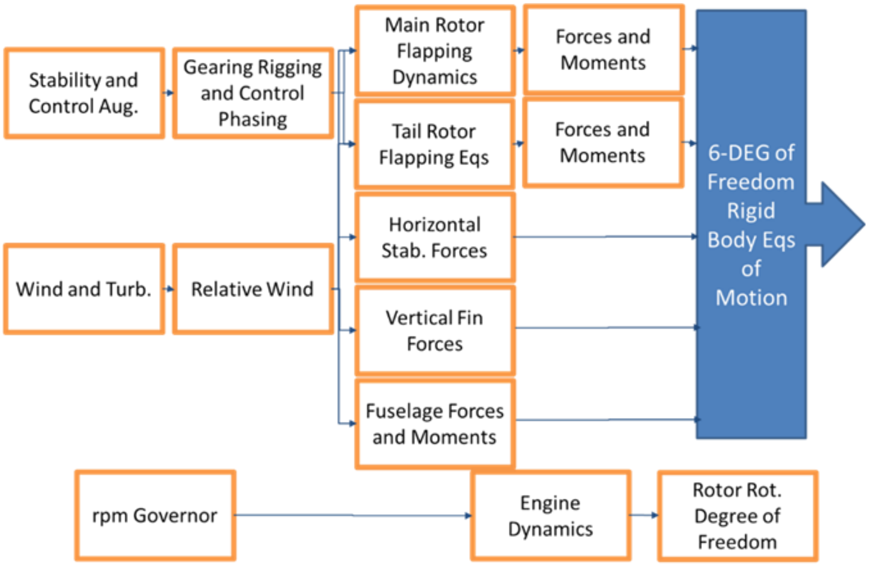


Figure 72. Components of the helicopter model

4.3.3.1 6DOF

The six degree-of-freedom model developed for the simulation takes in the forces and moment on the helicopter's center-of-gravity and solves a differential equation to determine the helicopter state as a result of the combined forces on the helicopter. This calculation shown in Figure 73 makes use of the weight of the helicopter, the inertia matrix, and the initial state in order to solve for the new state after a short time step. The moments imparted on the helicopter change its orientation and the forces imparted on the helicopter change its velocity.

$$\overline{F}_b = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = m\left(\dot{\overline{V}}_b + \overline{\omega} \times \overline{V}_b\right) \qquad \overline{M}_B = \begin{bmatrix} L \\ M \\ N \end{bmatrix} = I\dot{\overline{\omega}} + \overline{\omega} \times (I\overline{\omega})$$

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \equiv J^{-1} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = J \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & (\sin\phi\tan\theta) & (\cos\phi\tan\theta) \\ 0 & \cos\phi & -\sin\phi \\ 0 & \dfrac{\sin\phi}{\cos\theta} & \dfrac{\cos\phi}{\cos\theta} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

Figure 73. Six degree-of-freedom helicopter model equations

4.3.3.2 9DOF Rigid-body two-bladed teetering rotor model

The development of a 9 Degree-of-Freedom (DOF), rigid-body, two-bladed teetering rotor helicopter model was initiated in previous work by Robert Chen at NASA Ames (Chen, 1979). The major outcome of these reports is a reduced set of equations for modeling rotor flapping of a teetering rotor. In particular, the teetering rotor is modeled with a set of key parameters and assumptions such as:

- Flapping hinge restraint
- Effective hinge offset
- Blade lock number
- Pitch-flap coupling

To further simplify the model, the N-bladed teetering rotor model was reduced to a two-bladed teetering rotor. The primary result of this is the assumption that the hinge offset ratio and the flapping hinge offset are zero and the blade coning angle is a constant, leading to the simplification shown in Equations 14.

$$\epsilon = \dot{a_o} = \ddot{a_o} = 0 \qquad 14$$

The primary set of equations remains the same and are as in Equation 15.

$$\ddot{\bar{a}} + \grave{D}\dot{\bar{a}} + \acute{K}\bar{a} = \acute{f} \qquad 15$$

Where tip path plane is represented by the blade coning angle and the longitudinal and lateral first-harmonic flapping coefficients as in Equation 16.

$$\bar{a} = \begin{bmatrix} a_0 \\ a_1 \\ b_1 \end{bmatrix} \qquad 16$$

Therefore, the 9 DOF model consists of state variables corresponding to the helicopter position, velocity, orientation, and the tip path plane of the main rotor blade as follows:

$$x_e, y_e, z_e,\ u,\ v,\ w,\ \theta,\ \phi, \psi,\ a_0,\ a_1, b_1$$

First, the cyclic control coefficients are calculated for the helicopter constraints and the control parameters. These coefficients are used in the calculation of the main rotor forces and moments, along with the flapping coefficients. The tail rotor forces and moments, and the aerodynamics forces and moments on the fuselage are then calculated, and the model includes the rotations per minute or RPM governor. Each of these force and moment considerations are transformed to the center-of-gravity hub-body axis of the helicopter before being compiled and used to find the helicopter state with the 9 DOF model.

4.4 Implementation

The interaction between the cutter and the WSPS involves many different disciplines. The three main actors in modelling the physics of a wire strike are the helicopter, the wire, and the WSPS. The helicopter can be described dynamically with its inertias, translational and angular velocities, and orientation in three dimensions. The wire properties consist of its material properties, as well as the energy the wire stores in the form of elastic energy. Lastly, the WSPS key properties relate to the geometry of the blade used to ultimately cut the wire. In order to begin the task of modelling the interaction between wire cutter and wire, a simplistic initial scenario can be evaluated and then built upon with more factors. This initial scenario is one where the helicopter is moving straight towards the wire with a constant velocity. Given the basic properties of each of the three actors in the simulation, a determination for whether the wire is cut or not can be extracted.

4.4.1 Object oriented model

Within the various models of the wire cutter, the wire, and the helicopter, there is a multitude of parameters for each actor in the simulation as a whole. Due to this, an object-oriented programming approach was taken to organize the variables needed. The three main objects in the wire strike simulation are the helicopter, the wire, and the WSPS. The helicopter object holds all of the properties required for the rotorcraft simulation, and the functions required to complete the helicopter simulation. The wire object holds all of the properties of each wire, and the methods for calculating the resultant forces in the wire and the axial stiffness of the wire. Finally, the WSPS class holds the properties of the protection system and the function to calculate the force required to cut through the wire. The properties for each class are initialized in the scenario script, which calls upon three separate functions that initialize the properties in each class. These initialization functions all take in strings that describe the object being referenced. For example, the helicopter initialization function takes in the model *Bell 206* and fills in the appropriate properties to the helicopter class, and the same method is completed for the initialization of the wire. Currently, there is only one wire cutter being considered, that being the WSPS, but in future research, the wire cutter type would also be an input to the model. Once the different objects of the wire strike model are initialized, the wire strike simulation script can begin.

4.4.2 Wire cutting script

The first step of the wire strike simulation process is to determine the criteria for the wire being cut, which based on the described model constitutes a force requirement. The model discussed in the survey of literature defines and uses the shear area to determine what the required cutting force is for an object. However, the calculation of the shear area is based on a material with a rectangular cross section, while the wire has an approximately circular cross section. In this case, one of the methods in the wire strike protection system class uses the properties of the wire and the orientation of the helicopter to determine what this shear area would be for various scenarios. The calculation is considered by first visualizing cross sections of a cylinder taken at different angles, which can be either a circle or an ellipse. Then, the geometry of the wire cutter is considered, particularly the opening angle of the blades. Where the blades are tangent with the cross section of the ellipse indicates an important geometry for determining the area from the intersections of the blades to this intersection. The shear area will ultimately be the area of the triangle as shown in Figure 74, with the area of the cross section within this area subtracted. The method takes in the geometry of the WSPS, the orientation of the helicopter, and the diameter of the wire to determine the shear area, and ultimately the required cutting force.

Figure 74. Shear area between the WSPS and the wire

4.4.3 Wire dynamics scripts

The elastic modulus calculation is completed by a method in the wire class, which uses the properties, such as number of strands, wire diameter, wire material, etc., of the wire to calculate the estimation of the elastic modulus. The wire dynamics model for this study is also implemented differently than in Wu and Cao (Wu & Cao, 2016). While this study described the wire as a series of small, constrained cylinders, this level of intricacy led to a more complicated script with longer run times than desired for a M&S environment aimed at making quick analysis. In response to this challenge, the original model was adapted to make it more efficient while maintaining an acceptable level of accuracy. The complexity was reduced by removing the description of the wire as a combination of cylindrical elements and replacing it with estimations of the displacement and internal forces in the wire using the wire mechanics model from *Cable Structures* (Irvine, 1981). The results of this altered model were compared against the results of the full model as given in Irvine (Irvine, 1981) for the same inputs of a high-speed moving block as was used in the study by Wu and Cao (Wu & Cao, 2016).

The comparison of the impact and resultant forces can be seen in Figure 75, the comparison of the kink-wave angle is shown in Figure 76, and the comparison of the velocity and acceleration over the experiments is shown in Figure 77 (Xu, Dong, Zhang, & Zhang, 2014). While there are some clear differences between the validated results and the results of the less complex model, the trends of each case are ultimately similar. In addition, the most important consideration for determining if the wire is cut is the highest magnitude of force applied to the helicopter or the resultant force. Given that these forces peak at similar values to the validated results, the simplified model shows reasonable accuracy for this specific case. Given that this model has been validated separately, it is applicable to being adapted to the wire strike case, such that the high-speed block is replaced with the helicopter dynamics.

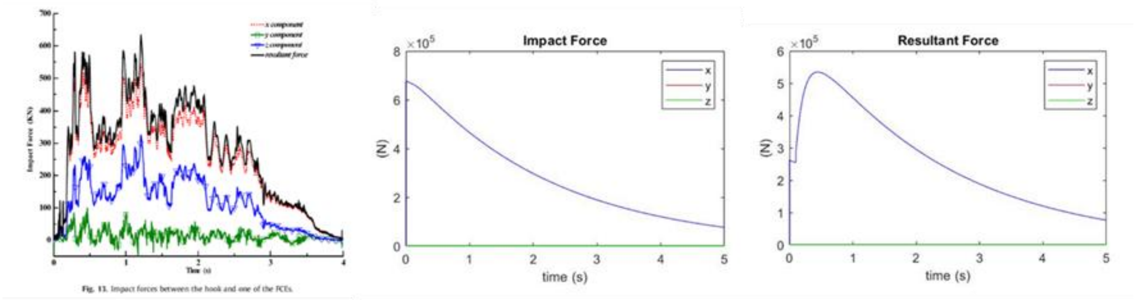


Figure 75. Comparison of validated impact force and calculated impact and resultant force

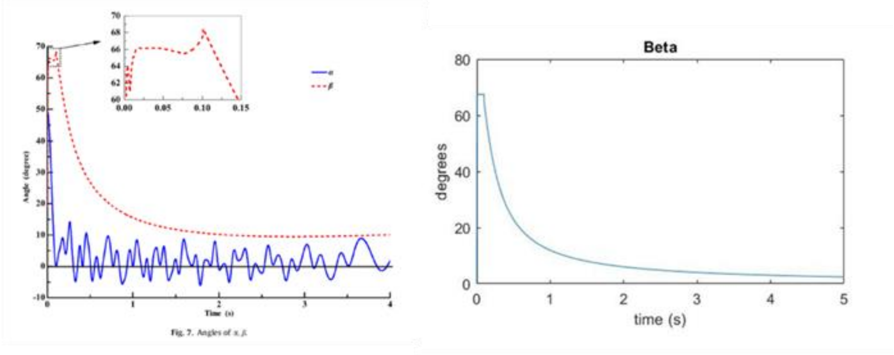


Figure 76. Comparison of validated wave angle and calculated wave angle

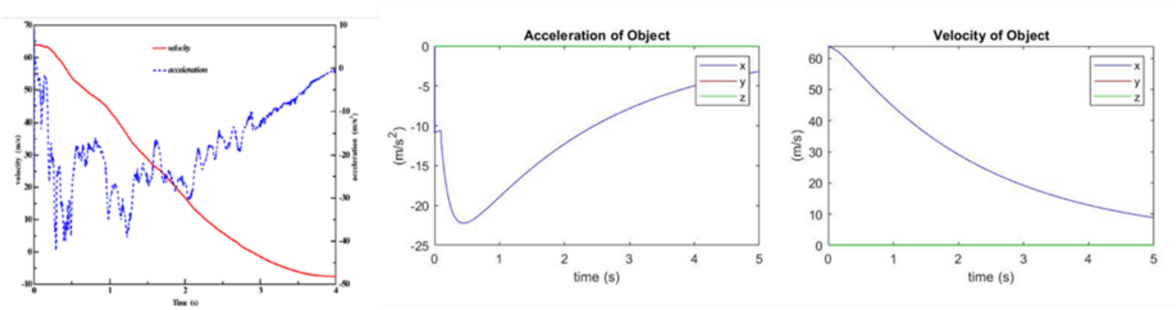


Figure 77. Comparison of validated acceleration and velocity over time and calculated acceleration and velocity

4.4.4 Helicopter model scripts

The goal of this work is to design and test wire cutter designs on helicopters and therefore, a helicopter model is needed inside the simulation environment. The helicopter dynamics were modeled in MATLAB with the same equations given in Section 4.3.3 on rotorcraft simulation. Initially, the main rotor and tail rotor forces were the only forces considered. However, upon testing the previously developed helicopter model, it was determined that more of the force considerations from the original model would need to be included for an accurate representation. The model was implemented such that there are separate scripts that calculate the forces and moments on the center of gravity of the helicopter for each major helicopter component. These scripts are for the main rotor forces and moments, the tail rotor forces and moments, the cyclic controls, and the empennage forces and moments. The helicopter object contains vectors of parameters required for calculations of the relevant forces and moments. The main rotor script also includes the calculations of the flapping dynamics variables by solving the differential equation of the flapping dynamics. Once these calculations are completed, the forces and moments on the helicopter are compiled along with the external forces and moments on the helicopter in a separate method in the helicopter class to update the force and moment parameters of the helicopter object.

Finally, the helicopter state is updated using a 9 DOF calculation of the helicopter dynamics. The forces and moments on each helicopter component are updated since they are partially dependent on the helicopter state, and the state is updated again. This cycle continues for the specified amount of simulation time. The validation of the rotorcraft simulation involved using the example helicopter parameters given for the AH-1G and inputting an initial state vector and control parameters corresponding to a trimmed helicopter state. The conditions for trim for the AH-1G were found in a reference from the helicopter model paper (Davis, 1974). The resulting forces, moments, and helicopter state over time can then be analyzed to determine if the behavior of the simulated rotorcraft is representative of being trimmed. For example, the lift force from the main rotor should be close to the weight of the helicopter, showing that the helicopter is not climbing or descending. Once this is the case, it can be determined that the scripts completing the rotorcraft simulation are working as intended. Upon implementation, the results of the simulation indicated that more of the full model would need to be included for reasonable accuracy. As seen in Figure 78, despite inputting trim conditions, the helicopter state does not seem to be converging. In order to stabilize the moments inputted to the helicopter, considerations for the empennage will be included in future additions to the helicopter model.

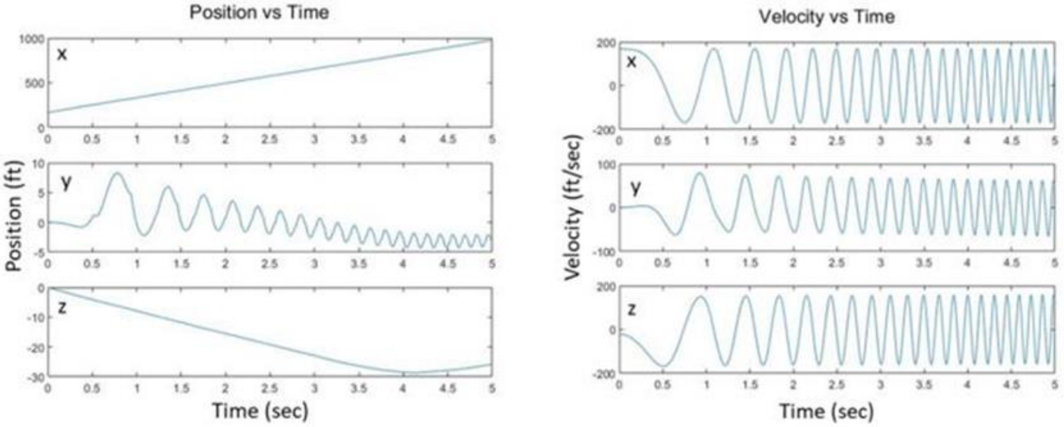
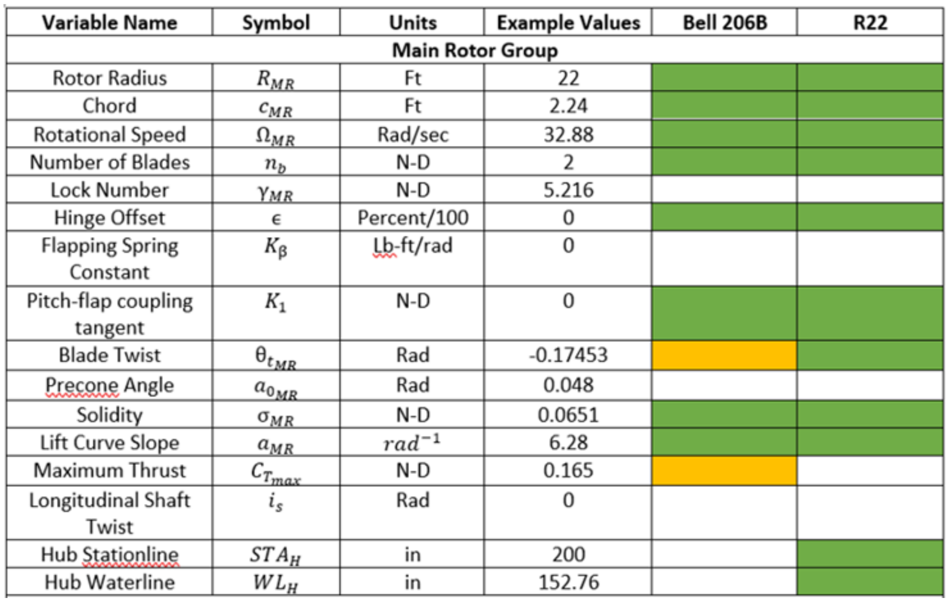


Figure 78. Simulated helicopter position and velocity given a force and moment input

4.4.5 Rotorcraft model data

One challenge to the implementation of the previously developed modelling and simulation environment for various helicopter types was the number of parameters required to complete the force and moment calculations. The helicopter model developed in this study provides all parameters needed for the AH-1G model, but for other helicopter models of interest, these parameters are not immediately available. From an analysis of rotorcraft wire strike accidents provided by the NTSB in their access database, we found that wire strike accident data is mainly available for some of the most popular helicopters, the *Bell 206* and the *Robinson R22*. Some parameters for these rotorcrafts were publicly available, and some further information was provided after contacting helicopter manufacturers. After completing an assessment of the information available to the team, the table shown in Figure 79 was developed. For the remaining parameters of the helicopters of interest, the team is determining which values can be assumed, and which values should be pursued for more accuracy.



Variable Name	Symbol	Units	Example Values	Bell 206B	R22
Main Rotor Group					
Rotor Radius	R_{MR}	Ft	22	🟩	🟩
Chord	c_{MR}	Ft	2.24	🟩	🟩
Rotational Speed	Ω_{MR}	Rad/sec	32.88	🟩	🟩
Number of Blades	n_b	N-D	2	🟩	🟩
Lock Number	γ_{MR}	N-D	5.216		
Hinge Offset	ϵ	Percent/100	0	🟩	🟩
Flapping Spring Constant	K_β	Lb-ft/rad	0		
Pitch-flap coupling tangent	K_1	N-D	0	🟩	🟩
Blade Twist	$\theta_{t_{MR}}$	Rad	-0.17453	🟨	🟩
Precone Angle	$a_{0_{MR}}$	Rad	0.048		
Solidity	σ_{MR}	N-D	0.0651	🟩	🟩
Lift Curve Slope	a_{MR}	rad^{-1}	6.28	🟩	🟩
Maximum Thrust	$C_{T_{max}}$	N-D	0.165	🟨	
Longitudinal Shaft Twist	i_s	Rad	0		
Hub Stationline	STA_H	in	200		🟩
Hub Waterline	WL_H	in	152.76		🟩

Figure 79. Main rotor parameters available for the Bell 206 and Robinson R22

4.4.6 Wire strike simulation

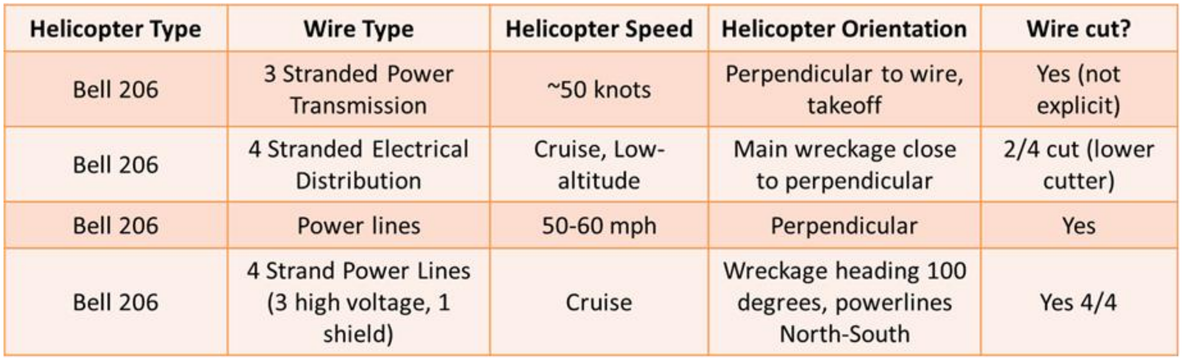
The wire strike simulation is the combination of the various methods described by the cutting mechanics, wire dynamics and mechanics, and helicopter dynamics models. The simulation scenario script initializes the main parameters of each object in the simulation, as well as the initial state of the helicopter. Another element of the scenario defines the number of wires the helicopter will have to go through, given that many powerlines have more than one wire to be cut. The wire strike simulation begins with the wire dynamics and mechanics model which determines the impact forces over an initial short time step and applies the resultant force as an external force and moment of force on the helicopter. The rotorcraft simulation is run for a short time step as well given the initial conditions and the external forces and moments, and a new helicopter state is determined. This new state determines the impacting forces and resulting force of the next time step, and the cycle continues until it is determined that the resultant force is greater than the required force to cut the wire, and the wire is cut.

4.5 Validation cases and testing

4.5.1 Wire strike cases in the NTSB database

While each of the models used to describe each discipline were validated individually, another challenge to the wire strike simulation was determining a way to compare the results of the full model to validated or verified results. There were two main sources of validation for the wire strike model as a whole. The first is the results of WSPS validation testing completed by the U.S.

Army as discussed in Section 4.1. One of the helicopters tested for compatibility with the WSPS is the AH-1S (Burrows L. T., 1982) which is a similar model to the AH-1G whose parameters are given in the example case of the helicopter model. These experimental cases provide an avenue of initial testing for the wire strike model, and when it comes to completing tests for various types of helicopters, the next source of validation is accident reports for helicopter wire strikes. As mentioned previously, some of the more popular helicopter models, the BELL 206 and the Robinson R22, have more available wire strike accident reports in the NTSB database, and because the research team for this study had the most information about these helicopter models in comparison to other models, accident reports for the Bell 206 and the Robinson R22 models were looked into exclusively. The resulting wire strike accident reports were searched individually for the following characteristics: the use of a WSPS, whether the wire was cut, the state of the helicopter just before impact, and the type of wire impacted. A few examples of reports with viable information can be seen in Figure 80. While all the reports did not have all of the required information, some information could be extrapolated and quantified. For example, a report may say the helicopter was cruising before the crash, so the speed of the helicopter can be estimated as that model's cruise speed. The Bell 206 is a medium weight helicopter with WSPS capabilities, so report data for this model can be used along with the experimental testing results for validation of the model. The R22, however, is a lightweight helicopter that does not have a WSPS model, so the reports for this model can be used to compare the results of the simulated performance of the WSPS on the R22 in a wire strike accident to the outcome of the actual reported accident without the WSPS.



Helicopter Type	Wire Type	Helicopter Speed	Helicopter Orientation	Wire cut?
Bell 206	3 Stranded Power Transmission	~50 knots	Perpendicular to wire, takeoff	Yes (not explicit)
Bell 206	4 Stranded Electrical Distribution	Cruise, Low-altitude	Main wreckage close to perpendicular	2/4 cut (lower cutter)
Bell 206	Power lines	50-60 mph	Perpendicular	Yes
Bell 206	4 Strand Power Lines (3 high voltage, 1 shield)	Cruise	Wreckage heading 100 degrees, powerlines North-South	Yes 4/4

Figure 80. Wire strike accident information from the NTSB

4.5.2 Results of the wire dynamics and cutting mechanics model

While the finalized helicopter model used for the simulation provides a more accurate representation of the dynamics of the impacting object at each time step, these simulation results have been obtained by treating the helicopter similar to the impacting object by excluding moments. Using this estimation, the wire strike event simulation was run for a case of the AH-1G experimental study (Burrows L. T., 1982), the OH-58A experimental study (Burrows L. T., 1982), and three wire strike incident cases from the accident reports created by the NTSB. The AH-1G case tested the WSPS for a pre-impact initial velocity of 40 knots, and the wire simulated was a 50-stranded communications cable with an estimated tensile strength of ten-thousand pounds per square inch (Burrows L. T., 1982). In this test, the wire was successfully severed. The results of simulating this case are shown in Figure 81, where the left graph represents the velocity of the helicopter during the impact, the middle graph is the resultant force, and the right graph is a closer view of the resultant force at the beginning of the impact. The horizontal purple line represents the force required to cut the wire. In this case the simulation correctly predicts that the wire will be severed, as the resultant force exceeds the required force at the beginning of the impact.

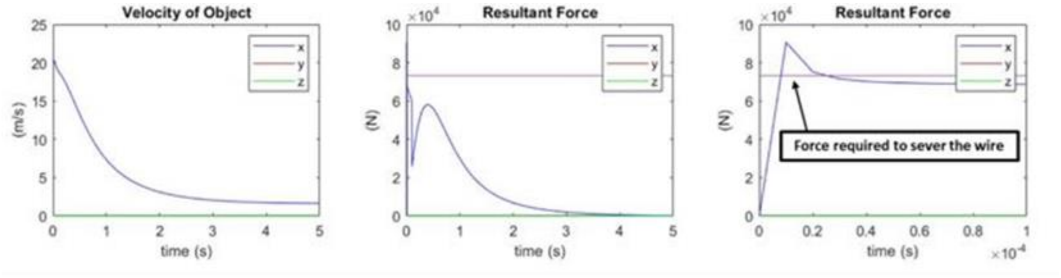


Figure 81. AH-1G simulated wire strike event

The next case considered for validation was one of the experimental tests on the OH-58A. The same wire is being considered, but the helicopter weight is less, and the pre-impact velocity is 37 knots rather than 40. In the experiment, the wire was severed, and the simulated results shown in Figure 82 align with the verified results as well (Burrows L. T., 1980). In addition, because the helicopter weight is lower, the decrease in helicopter velocity, seen in Figure 82 on the left, is steeper than for the AH-1G.

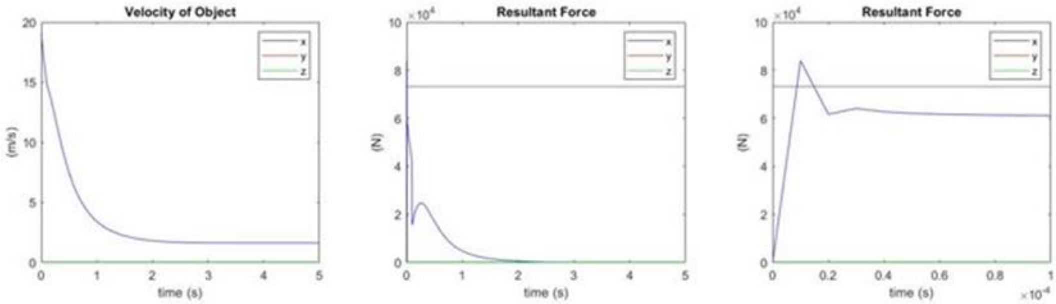
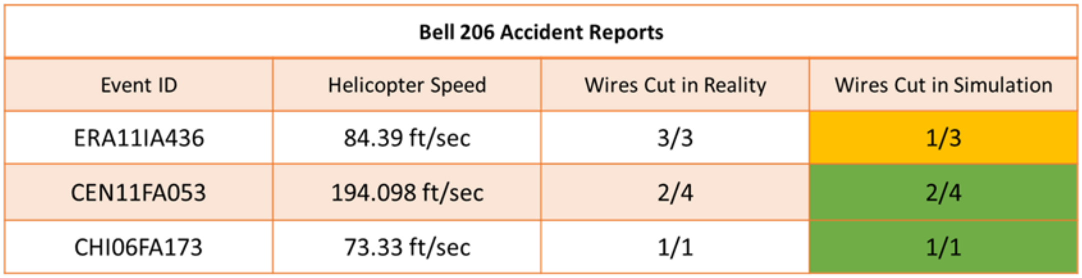


Figure 82. OH-58A wire strike event simulation

The next few validation cases completed were for accidents involving a WSPS equipped model of the Bell 206. Figure 83 provides a summary of the three cases and the simulated results. The wire strike event simulation was accurate for two out of the three cases considered, including one involving multiple wires. However, the incorrect result where the simulation predicted that one wire would be cut when all three wires were cut indicates that the model needs more refinement for strikes in which multiple wires are involved.



Bell 206 Accident Reports			
Event ID	Helicopter Speed	Wires Cut in Reality	Wires Cut in Simulation
ERA11IA436	84.39 ft/sec	3/3	1/3
CEN11FA053	194.098 ft/sec	2/4	2/4
CHI06FA173	73.33 ft/sec	1/1	1/1

Figure 83. Results of wire strike simulation for three representative NTSB accidents

Finally, a few cases for wire strike accidents involving the R22 without a WSPS were simulated as well, as one of the goals of this study is to determine how well the passive (static) wire cutters would sever wires on lightweight helicopters.

The results of this simulation are shown in Figure 84, and the resultant forces shown in the center and right of Figure 84 indicate that the R22 performs better than the heavier helicopters in severing wires. This result did not align with the initial intuition regarding lightweight helicopters in wire strike situations, and an investigation into the implementation of the model highlighted the fact that the impact forces as calculated in **Error! Reference source not found.** a re not dependent on the weight of the object on the first step of the simulation, but the force is mainly dependent on the velocity. In this particular example case, the initial velocity of the R22 is much faster than most of the previous validation cases. In the example case shown in the bottom of Figure 84, the R22 still cuts the wire, but not as easily. This may indicate that helicopter velocity is a more influential variable when it comes to wire cutting than helicopter weight. With regards to the energy of the helicopter, changes in velocity are more influential to the kinetic energy than changes in mass as well. In addition, the state of the R22, as shown by the velocity in Figure 84 is more severely impacted than the state of heavier helicopters. This may indicate that lighter helicopters are able to cut wires at high speeds but are not controllable after impact.

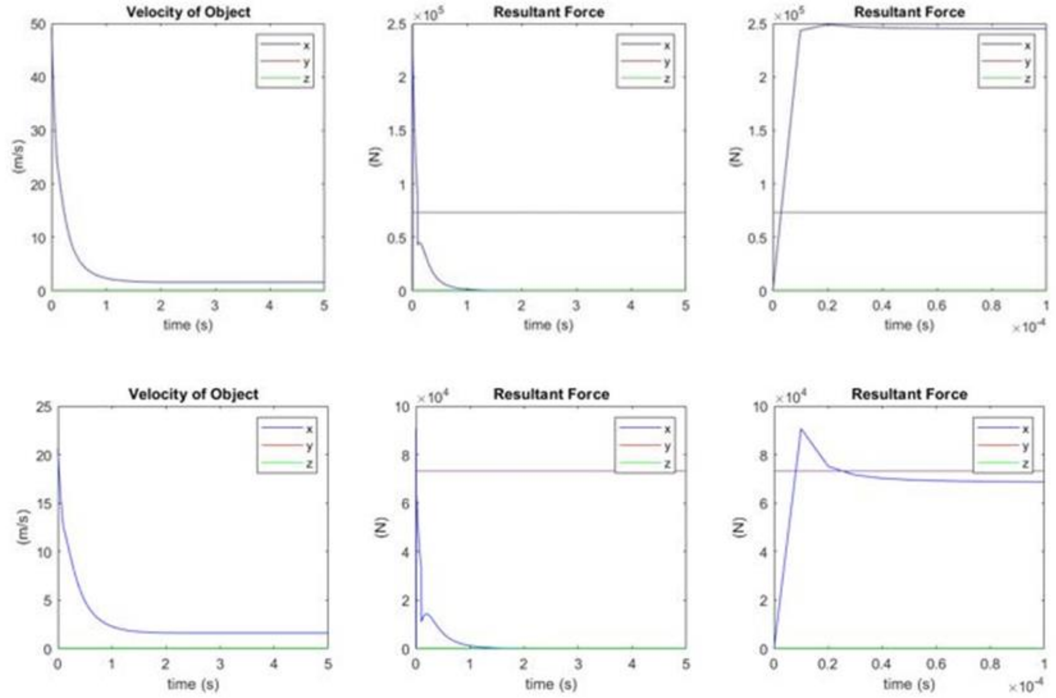


Figure 84. R22 simulated wire strike event for high and low initial velocities

4.6 Conclusions on physics of wire cutting

The goal of this research was to create a physics-based simulation of the impact between a helicopter and a wire, and determine whether a wire could be severed in the interaction. The authors identified the main disciplines required to achieve this goal, those being cutting mechanics, wire dynamics, and helicopter dynamics, and also completed a review of related literature to understand existing models in various industries and how they could be applied to the wire strike case. Finally, the most applicable models were chosen and modeled using object-oriented programming in MATLAB. A combination of an estimation for the force required to sever through the wire and a simplified model for the wire behavior after an impact constituted the majority of the wire strike event simulation. The simplified wire dynamics model was first calibrated and validated against the results of the full model it was derived from, and it was shown that the simplified model captured the most important aspects of the wire dynamics well, those being the impacting forces and the forces imparted back onto the impacting object. The implementation of a rotorcraft simulation was also begun as a part of this study, in preparation for further investigating the behavior of the helicopter after the impact.

Ultimately, the initial result of the research study was to validate the implemented simulation consisting of the cutting force estimation and the simplified wire dynamics model. The wire strike event simulation was able to simulate a number of events described in the WSPS experimental studies (Burrows L. T., 1982) (Burrows L. T., 1980), and from wire strike accident reports created by the NTSB. The results of these simulations were in accordance with verified results for strikes involving one wire and were less accurate compared to verified results for strikes involving multiple wires. Finally, an initial investigation into the simulated results for accident reports involving a lightweight helicopter, the R22, was completed, and areas of interest for the simulation of the lightweight case was revealed for future work. Overall, the work in this study provides a foundation for more complex physics-based wire strike event simulations.

Future work could build upon the initial foundation completed to analyze different scenarios and combinations of object properties. This could include the completion and implementation of the full rotorcraft simulation, gathering parameter data for more helicopters, and completing testing for more accident reports where the helicopter is impacting at a variety of speeds and orientations. Changing the location of the wire cutter on the helicopter will also change the resultant helicopter behavior after the impact, and this could also be included in the future versions of the wire strike event simulation to account for the lower cutter of the WSPS, as well as other placements for protecting different parts of the helicopter. In addition to determining the efficacy of the currently available wire strike protection systems on lightweight helicopters, another area for future work would be to design and simulate new wire cutter designs such as active wire cutter systems (moving mechanism to slice the wire). These future designs could be modeled with the existing wire cutter element by expanding upon the methods of the wire cutter object to include a new physics-based estimation of the criteria for a successful cut. The M&S environment could then be used to make quick analysis on what active designs are most efficient for wire strike incidents on lightweight helicopters and approach the goal of developing an effective wire strike safety mechanism for lightweight helicopters.

5 Investigation of sensor information fusion for wire detection and wire database improvement

The objective of this research was to investigate the use of onboard sensor data to verify or update the wire database during flight. This could add new information to the wire database in areas of low confidence or where there are discontinuous sections, and then could continue validating or updating the three-dimensional location of all power lines. This research creates a methodology and develops a software environment to begin improving the wire database, which would eventually provide insight on onboard sensor requirements and improve pilot response for wire strike prevention. The following methodology assumes that a wire database exists (as obtained from the work presented in Section 2) and that onboard sensing is, or will be, available to verify wire databases during or after flights.

The wire database and experimental radar data for a specific location are assumed to be aligned. The assumptions of the wire database and radar sensor uncertainty, as well as the global reference frames for each data source are required for the data fusion algorithm. The methodology leverages Bayes Theorem to predict whether the local area is occupied with wires or not and the output generated is a distribution of the likelihood of the area being occupied. This update process is used to improve the wire database.

The data fusion pipeline is condensed into a four-step process that combines multiple algorithms. First, an occupancy grid is prepared for experimentation. Second, the wire database is mapped into the occupancy grid with confidence bounds and a resolution as provided by the experiment settings. Third, the radar data is processed through a sensor update using Bayes Theorem and occupancy grid update equations. Last, the wire database is modified by fitting the occupancy grid data utilizing the new data from the sensors, to a new and improved line representing the 2D position of the wire in the global coordinate system. A series of experiments are prepared to investigate how data fusion can improve the detection of wires and lead to better wire strike prevention systems for rotorcraft safety.

5.1 Data fusion introduction and background

The wire database obtained from the process described in Section 2 is assumed to exist in an area of interest where wire locations are important for rotorcraft safety. The current methodology allows for maps to be generated in regions with sufficiently high-resolution satellite imagery and street view imagery using trained deep neural networks. Currently, results are available in about 60 square miles of Westchester County in the state of New York, and can be exported as a shapefile, a data structure that can be imported into any GIS software.

Current results are promising, but there still exists false positives and false negatives, and the predicted areas of wires require a buffer of uncertainty of seven meters on both sides. Improved training and human labeling could reduce these, but without the ability to validate the entire wire map, concerns will remain. Therefore, additional sensor measurements are necessary to make conclusions on the *wire/no-wire* indication for a pilot. This could be applied in real-time by rotorcraft equipped with sensors or done offline through mapping missions by unmanned aerial vehicles (UAVs) or rotorcraft. While the requirements for computational limits and sensor noise differ, the offline and online cases are both considered for this initial investigation.

Work by Honeywell and others have investigated onboard detection and fusion for powerlines or other wires (Berthold, 2019) (Fryskowska, 2019) (Goshi, Darren and Case, Timothy and McKitterick, John and Bui, Long, 2012) (Ma, Goshi, Shih, & Sun, 2011) (Ma Q. a.-T., 2015). However, none have taken advantage of an offline, computed wire database. This raises the question, "can we leverage sensor detections with the wire database, and what does this mean for helicopter wire strike prevention?"

5.1.1 Literature of powerline or wire detection and fusion

Many techniques have been leveraged for fusing different data or sensor sources, with many relying on a probabilistic framework to provide inference using Bayesian theories. Previous work by Ma et al. (Ma Q. , Goshi, Shih, & Sun, 2011; Ma Q. , Goshi, Bui, & Sun, 2015) demonstrated a tracking algorithm for powerlines using millimeter-wave radar data streams. The approach was to threshold the radar data, detect lines with the Hough Transform algorithm, and classify lines as wires with an SVM classifier. The contribution of the work primarily comes from the use of Bragg features in the SVM decision space, and the use of a cascaded particle filter. The Bragg pattern is the quasi-periodicity of cable lines that have periodic peaks and through the SVM acts as an operator to *filter out* anything that is non-cable. The success of the methodology is demonstrated onboard a helicopter during flight tests around power lines. The focus of this work was on the millimeter-wave radio onboard the helicopter for flight testing and the requirements for particle filters run in real-time onboard the vehicle. Similar work has also shown that the Passice Millimeter Wave (PMMW) sensor is the most capable for power line imaging in part because of the radar to isolate the cable due to the diffraction of electromagnetic wave. While sensors such as the PMMW may be of interest in future work, the current research is less interested in the real-time tracking of the power lines and focused on the requirements for fusing with already known, but uncertain, datasets like a wire database.

Alternative methods exist for using probabilistic frameworks to match or cluster data, such as a database. Techniques such as data association or mapping have been used to build road networks or track objects. The map-matching method has been used for determining paths along a known road network from imprecise location measurement samples (Eisner, Funke, Herbst, Spillner, & Storandt, 2011). In addition, Probability Hypothesis density filters have been used for similar applications, in part because “the main advantages are that it avoids the detection, the data association and the track handling problems in conventional multiple-target tracking, and that it gives a parsimonious representation of the map in contrast to grid-based methods” (Lundquist, 2010).

The fundamental question of this research is whether an area of space around the rotorcraft contains a wire or not. Whether through an offline database or an online sensor, there must be some discretization of the space and some method for inferring a wire exists. This introduces two techniques that build from first principles theory in ML, Bayesian filtering and occupancy grids.

5.1.2 Bayesian filtering

To understand Bayesian filtering, let us assume we have predictions of the wires from the wire database. This will be considered a discrete prediction, where a wire exists or does not at a given point in space. Let us take this to be a random variable A, and assume we have a prior distribution of the parameters of our initial estimation, or *prior* denoted P(A). Now, let us choose a statistical model that reflects what we believe the value of A is given those parameters, which we will call our beliefs. The value representing these *beliefs* could be a continuous value, like the distance an object is from a sensor, or a discrete value, like whether an object exists or not.

By abstracting the prediction of the variable to parameters of a known model, the value can be solved using data or simulation. Examples of parametrized models include statistical distributions such as the Gaussian distribution or the binomial distribution.

We wish to determine the posterior, which requires the use of Bayes Theorem which is described by the relationship between the prior, $P(A)$, the likelihood, $P(B|A)$ and the posterior, $P(B)$, as described in Equation 17.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \qquad 17$$

Equation 17 allows for inference on whether an event has occurred while conditioned on one or more other events. For spatial and temporal constraints, assuming a Boolean (true or false) event, the occupancy grid can be used, as explained in detail by Thrun (Thrun, n.d.). This approach maps the generic probabilities to parameters that can be predicted or measured. The parameters include the probability that a point in space is occupied by a wire, the probability that the radar sensor detects a wire, and the respective conditional probabilities. The formulation of the problem could be introduced over continuous 3D space. However, the computational complexity can be reduced by defining a discrete set of points in the local 3D space using an occupancy grid, as shown in Equation 18.

$$p(occupied|radar)_{(x,y,z)} = \frac{p(radar|occupied)p(occupied)}{p(radar)}_{(x,y,z)} \tag{18}$$

5.1.3 Occupancy grid

Let us consider the probability that the space around (x,y,z) is occupied. This is a probability ranging from zero to one. If these predictions are assumed to occur in discrete points in time, and are independent from one another, then the predictions over time result in a series of multiplications. For a large number of discretized grid points, this multiplication can become computationally heavy. Therefore, an alternative approach can be taken to reduce the computational load by reformulating the Bayes equation with log-odds.

To introduce the log-odds formulation, let us consider the random variable *A* as the occupancy chance, and then the odds function as is detailed in Equation 19. The log odds are used to improve computational requirements by shifting a multiplication problem to an addition problem.

$$\begin{gathered} o(A|B) = \frac{p(A|B)}{p(\neg A|B)} \\ \lambda(B|A) = \frac{p(B|A)}{p(B|\neg A)} \\ o(A|B) = \lambda(B|A) * o(A) \end{gathered} \tag{19}$$

The statistical update in Equation 18 requires a multiplication for every cell in the discretized map. Instead, the equation is updated with a *log* function to update to an addition, as described in Equation 20.

$$\begin{gathered} \log\big(o(A|B)\big) = \log\big(\lambda(B|A) * o(A)\big) \\ \log\big(o(A|B)\big) = \log\big(\lambda(B|A)\big) + \log\big(o(A)\big) \end{gathered} \tag{20}$$

The measurements are accumulated over time as shown in Equation 21. The recursive method avoids calculations as explained in detail in Burgard et.al (Burgard, et al., n.d.).

$$l(A)_t = \log\big(o(A|B)\big)_t = \log\big(\lambda(B|A)\big)_{1:t} + \log\big(o(A)\big)_t \qquad 21$$

$$l(A)_t = l(A)_{t-1} + \log(o(A|B)_t) - l(A)_0$$

Then, to solve back for the probability, we use Equation 22.

$$o(A)_t = 1 - (\exp(l(A)_t^{-1} + 1)^{-1} \qquad 22$$

In Equation 21 and Equation 22, variable *A* represents whether a wire is occupying an area of space, and variable *B* represents whether the radar data indicates that the space is occupied. This must be repeated for each grid position, for instance position *(i,j,k)* if in 3D.

5.1.4 Detecting power lines

Updating an occupancy grid requires a backward sensor model to include $\log\big(o(A|B)\big)$. Therefore, fundamental approaches to detect lines in 2D and 3D data are investigated. Two important methods for detecting lines or other geometric objects are Random Sample Consensus (RANSAC) and Hough transform. The Hough transform shown in Figure 85 leverages the polar representation of the parameter space for a model of interest, in our case a line $y = mx + b$. The parameter space in polar form can be written as $x cos\theta + y sin\theta = \rho$ thus defining the 2D Hough space with ρ and θ. This technique is robust to outliers and efficient, however, it can be sensitive to noise and requires parameter tuning for the ‘sweet spot’.

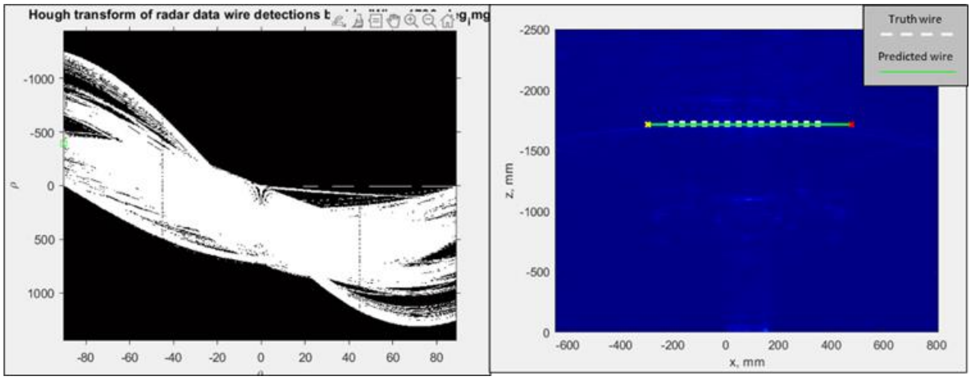


Figure 85. Hough transform

Random Sample Consensus, otherwise known as RANSAC, is a line fitting technique that leverages the model of interest like Hough transform. However, the objective is to fit the best set of inliers to a predefined threshold. This also provides robustness to outliers; however, the computational time can grow quickly with many outliers and not a clear best fit. An example of a multi-line RANSAC fit is shown in Figure 86. The pros and cons for each method differ and both are beneficial in different scenarios. Therefore, this work looks to leverage both methods as outlined in the methodology.

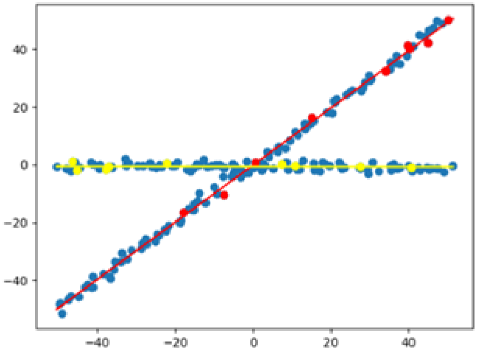


Figure 86. Multi-line RANSAC

5.2 Data fusion methodology

The process for fusing multiple data sources requires leveraging the math and algorithms detailed in the previous section. Furthermore, they must be formulated, parameterized, and programmed for the datasets and experiments that apply to this research. This culminates in the development of a software environment for wire database and radar detection fusion with a data pipeline for existing data. Fundamentally, the framework leverages Bayes theorem updates and an occupancy grid map for fusion. A Bragg pattern-based support vector machine is trained for wire classification and Hough transform is used for wire detection from radar measurement updates. Data handling requires geospatial sampling, rasterization, and spatial transformations to convert the wire database to the local occupancy map. The RANSAC wire line detection is used for reforming wire lines shapes to update the database. Experiments are run with the wire database from Westchester County and indoor radar data from Iowa State University. A high-level graphic of the data and fusion pipeline is shown in Figure 87.

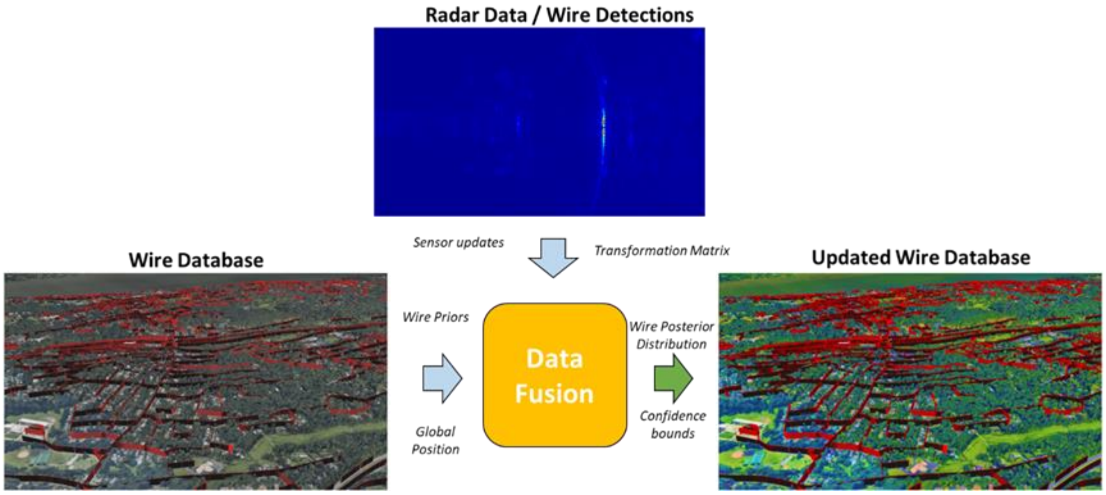


Figure 87. Data fusion pipeline

5.2.1 Prior information from wire database

The wire database from Westchester County, New York is processed to produce the priors in the occupancy grid. Figure 88 shows the truth and prediction lines, which as vector data must be rasterized and transformed into the occupancy grid coordinate system.

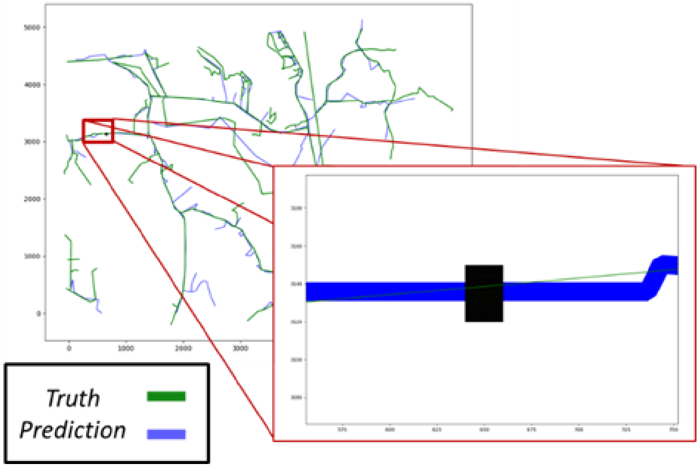


Figure 88. Wire database truth and prediction in Westchester County, NY

The wire database is assumed to have Gaussian uncertainty around the 2-dimension line stored in the wire database. Therefore, a buffer is added to the vector object and a Gaussian distribution is used when discretizing the predictions into the local grid.

5.2.2 Wire detection from radar data

Raw radar data from Iowa State University (ISU) provides insight into where a wire likely is from visual inspection. However, to provide accurate updates to the wire database, we need confidence in the location of the wire consistently and autonomously from the data. Therefore, the process shown here is used. First, the raw data is cleaned up by bounding the absolute values by a threshold selected from experimentation. Next, insight is taken from the Honeywell work shown earlier, and the combination of the Hough Line detection algorithm and Bragg SVM classification are used to predict the powerline in the radar image. Next, using the Bayes theorem and the occupancy grid, the top 10% of wire predictions are transformed into a 2D Gaussian distribution and discretized to the resolution of the occupancy grid. The pipeline for this process is shown in Figure 89.

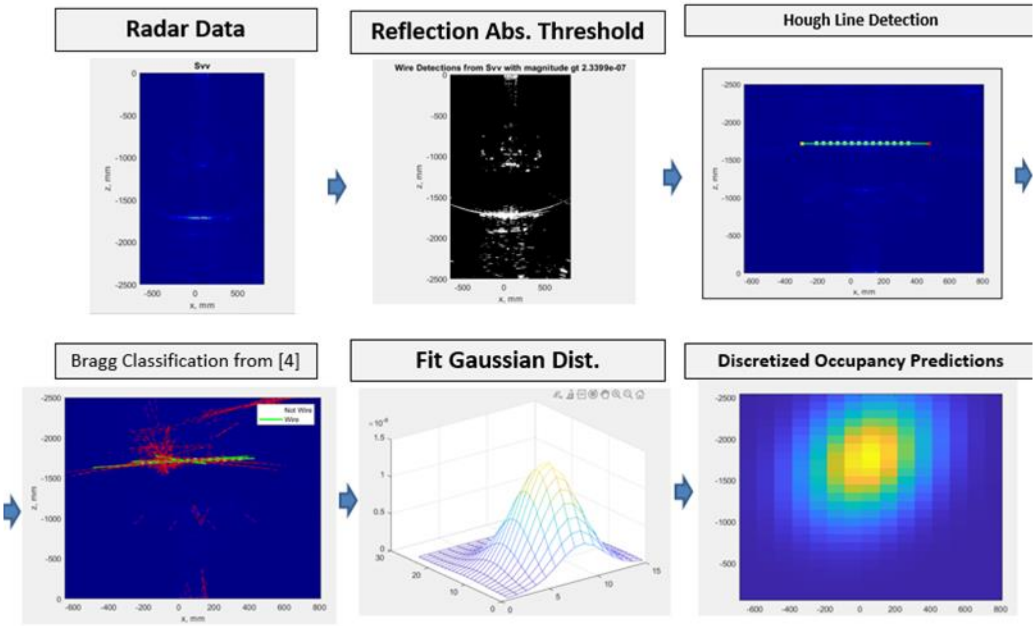


Figure 89. Wire detection methodology

5.2.2.1 Experimental data

The radar data was collected by ISU and a short summary is provided here for application to the data fusion study. An experimental setup to mimic the detection of powerlines from a helicopter using a single radar receiver/transmitter was considered as depicted in Figure 90.

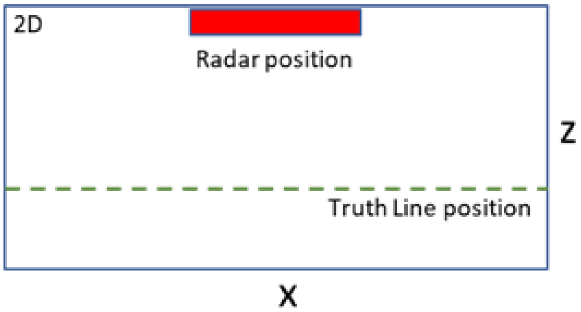


Figure 90. Experimental setup by Iowa State University

The sensor in use is a Ka-band (16.5 – 40 GHz) radar receiver and transmitter. A radar uses radio waves that are emitted and detected, like an echo, to determine if there are objects in the area of interest. More specifically, the data that is received by the radar is a complex number representing the reflection coefficient contrast, which is the relative magnitude and phase ratio between adjacent elements in the measurement space.

The ISU setup produces four channels for the four different polarization configurations. The polarization configurations depend on the transmission polarization and receiving polarization. The four channels can be seen in Figure 91 with the classification indicated by the *v*, for vertical, or *h*, for horizontal, and the sequential combination of the two.

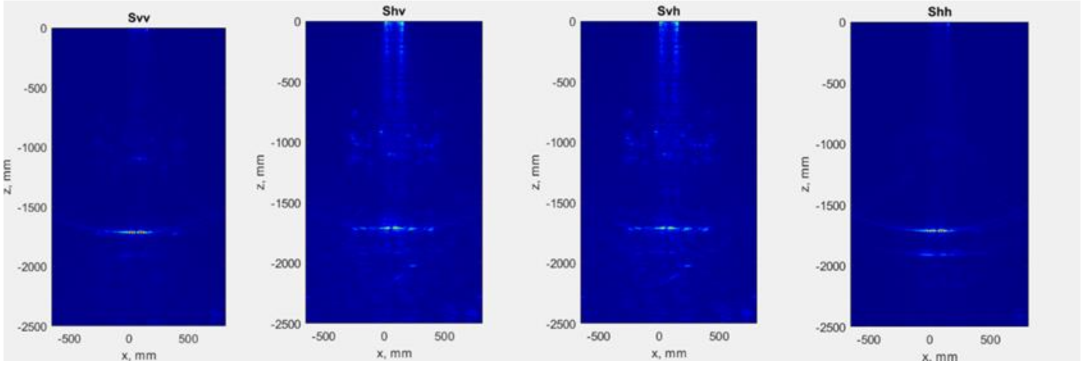


Figure 91. Absolute value of the real component of radar data in Cartesian coordinates for four different polarizations

An assumption was made that wires will *scatter* more than other objects for the current experiments. Therefore, estimates were made on the threshold -level of the magnitude that indicates a wire and leaves the uncertainty as an input parameter.

$$P(wire|radar, detection) = P(\theta) \sim N(\mu, \sigma^2) \qquad 23$$

5.2.2.2 Radar Noise

The noise crease *artifacts* in the images are most likely from other objects near the experiment. Therefore, a thresholding value was investigated to minimize the noise, while not losing information about the wire. The reflectance value threshold was selected to maximize the information for wire detection performance and an experiment investigating this value is found in Section 5.3.1.1. In Figure 92, the color white indicates a high enough reflectance to be a wire.

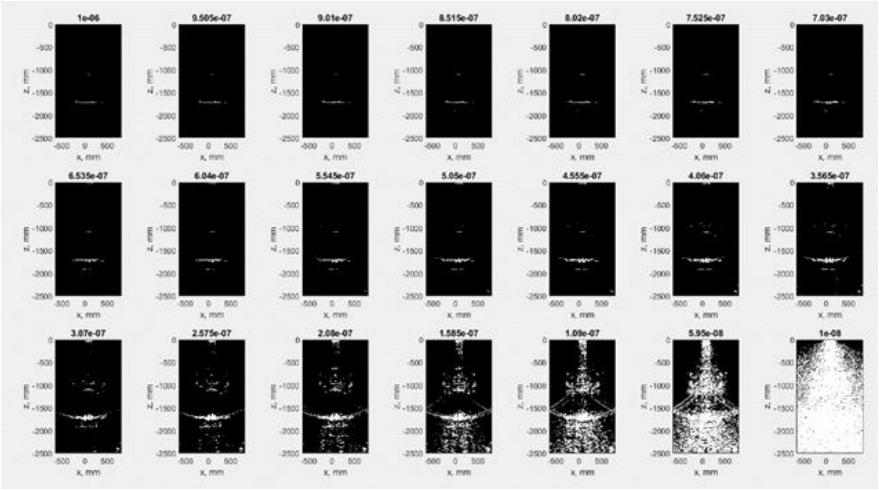


Figure 92. Radar data reflectance values when applying binary threshold (white pixels indicate a wire, black pixels indicate no wire)

5.2.2.3 Hough line detection and support vector machine Bragg Classification

The Hough transform is used to detect lines in the wire. For example, three angles of the data were reviewed to evaluate the performance, assuming the parameters can be tuned with the knowledge of the wire location. Experimental data of the line at 0 degrees rotation and the resulting angular and linear errors are shown in Figure 93.

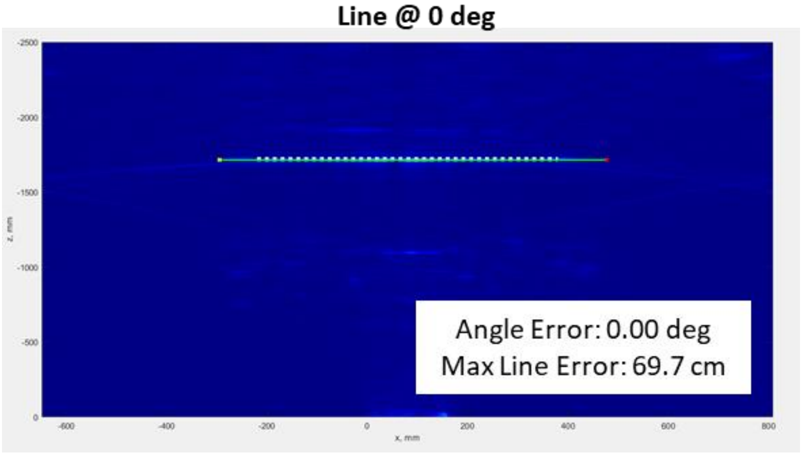


Figure 93. Experimental data at 0 degrees with Hough Transform fit results

However, without the truth data it is not clear which of the lines detected by Hough Transform is the wire. Therefore, techniques (Ma, Goshi, Shih, & Sun, 2011) were leveraged to improve the detection. Ma et al. (Ma, Goshi, Shih, & Sun, 2011) proposed a framework using Hough Transform and SVM, where the features are selected to characterize the Bragg Pattern. The Bragg features are detailed extensively in the previous work, but the first four features of the set include basic statistics of the data: the mean, max, 95% percentile, and 68% percentile. Other features relate the distribution of maximum values in the spatial data and the frequency of pixel magnitude changes. The assumption is that the Bragg features will indicate patterns in radar data with wires more than that without.

A Support Vector Machine (SVM) is a linear classifier often used for high-dimensional datasets. This is because the classifier acts as a separating hyperplane in high-dimensional space and leverages computationally efficient method technique that uses a subset of the data called support vectors. The simplest form of a SVM in 2D with a separating margin between the two classes (positive and negative) is seen in Figure 94. An optimization problem can be configured to solve for a hyperplane that separates the two classes. Furthermore, the SVM can handle nonlinearities in the high-dimensional space by using nonlinear kernel functions, for instance a Gaussian Radial Basis Function.

The optimization problem is discussed in more detail by Bishop (Bishop, 2006), but is introduced here for reference. A common practice is the transformation into the dual problem, which folds the linear classifier constraint into the objective function. The resulting formulation is a constrained quadratic programming problem that is nice and convex.

$$max\ L(w,\alpha,\beta) = \sum_{i}^{m} \alpha_i - \frac{1}{2}\sum_{i,j}^{m} \alpha_i \alpha_j y^i y^j \left({x^i}^T x^j\right)$$

$$s.t \quad \alpha_i \geq 0 \quad \forall \;\; i = 1, \dots, m$$

$$\sum_{i}^{m} \alpha_i y^i = 0 \tag{24}$$

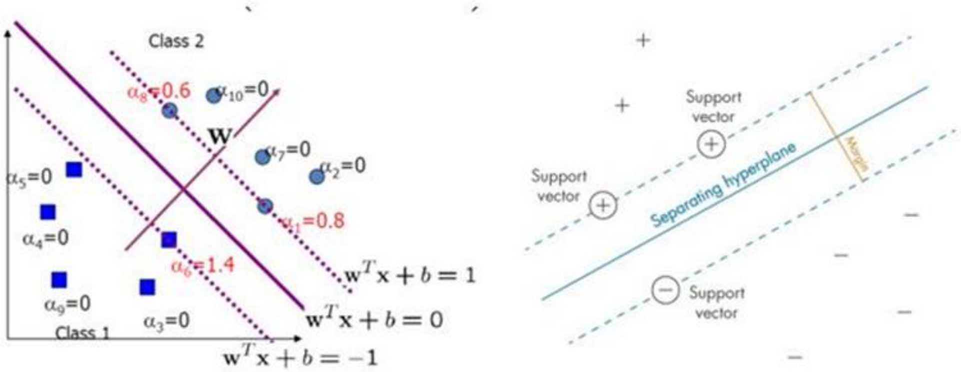


Figure 94. Vector machine graphic

The full pipeline for the wire detection phase is seen in Figure 95. The final step involves converting the wire classifications into a 2D gaussian with a maximum probability P'_{max} that is assumed from the radar sensing confidence (the likelihood in Bayes equation). The grid is discretized at the resolution of the occupancy. If the occupancy grid cells are modified to be larger than the sampled resolution, then a decision must be made on how to resample the data, whether through an average or max function.

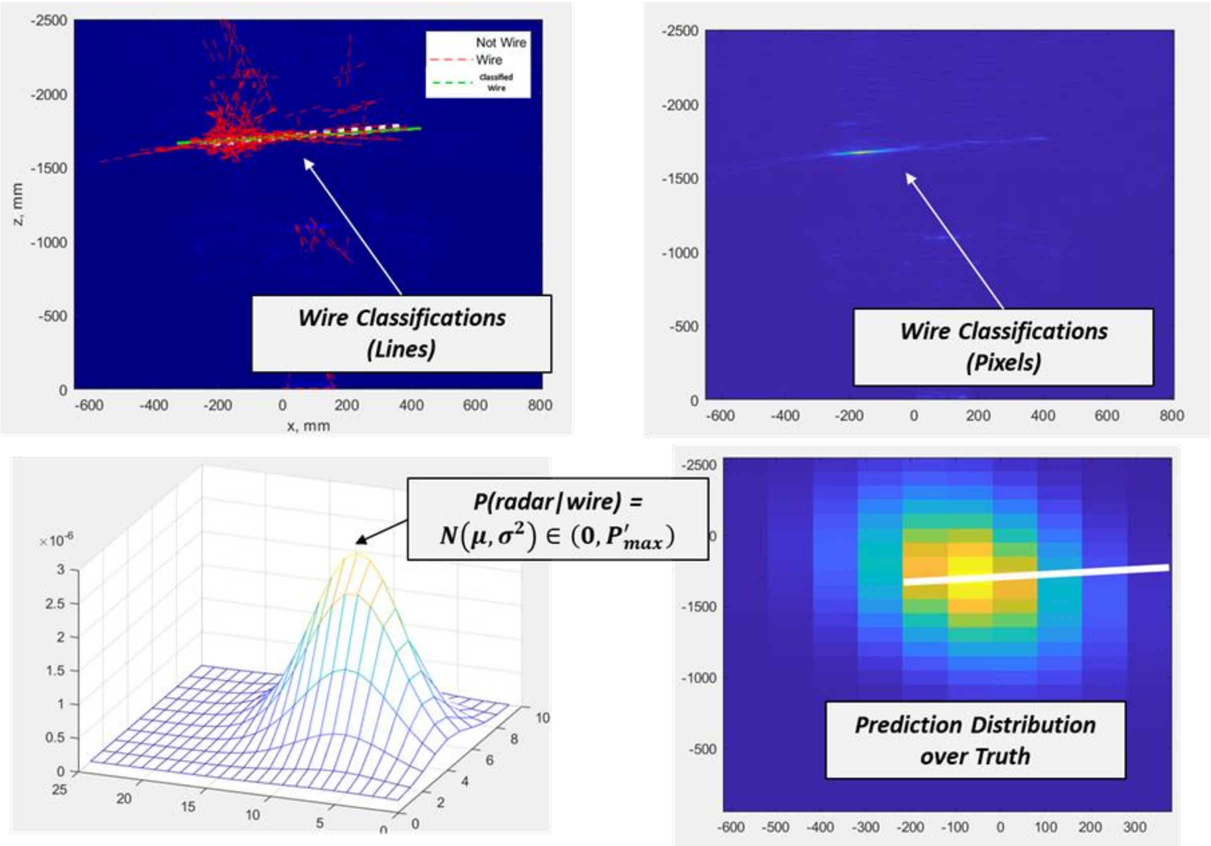


Figure 95. Wire detection pipeline of data

5.2.3 Occupancy grid mapping

Next, the radar detections are processed using Bayesian updates to update the occupancy grid. The grid is generated with a desired resolution, for example ½ meter, and utilized log-odds updates for more efficient calculations, where multiplication becomes addition shown on the right. An example of the setup is shown in Figure 96, with the wire location, vehicle position, and the 2D sensor measurements. On the right we see a 2D slice with the probability of a wire shown.

A simple case can be used to demonstrate the experiments. Consider the environment with a single wire in the environment, where there exists a prior knowledge of where the wire may exist. Everywhere else in the map is not fully certain, using a 25% prior. A single radar sensor then receives an update, with the assumption that sensor provides the 3D position with a confidence of 75%. Samples create other potentially occupied cells (~40%) but using a line fitting algorithm like RANSAC forms a confident line in 2D space for the wire.

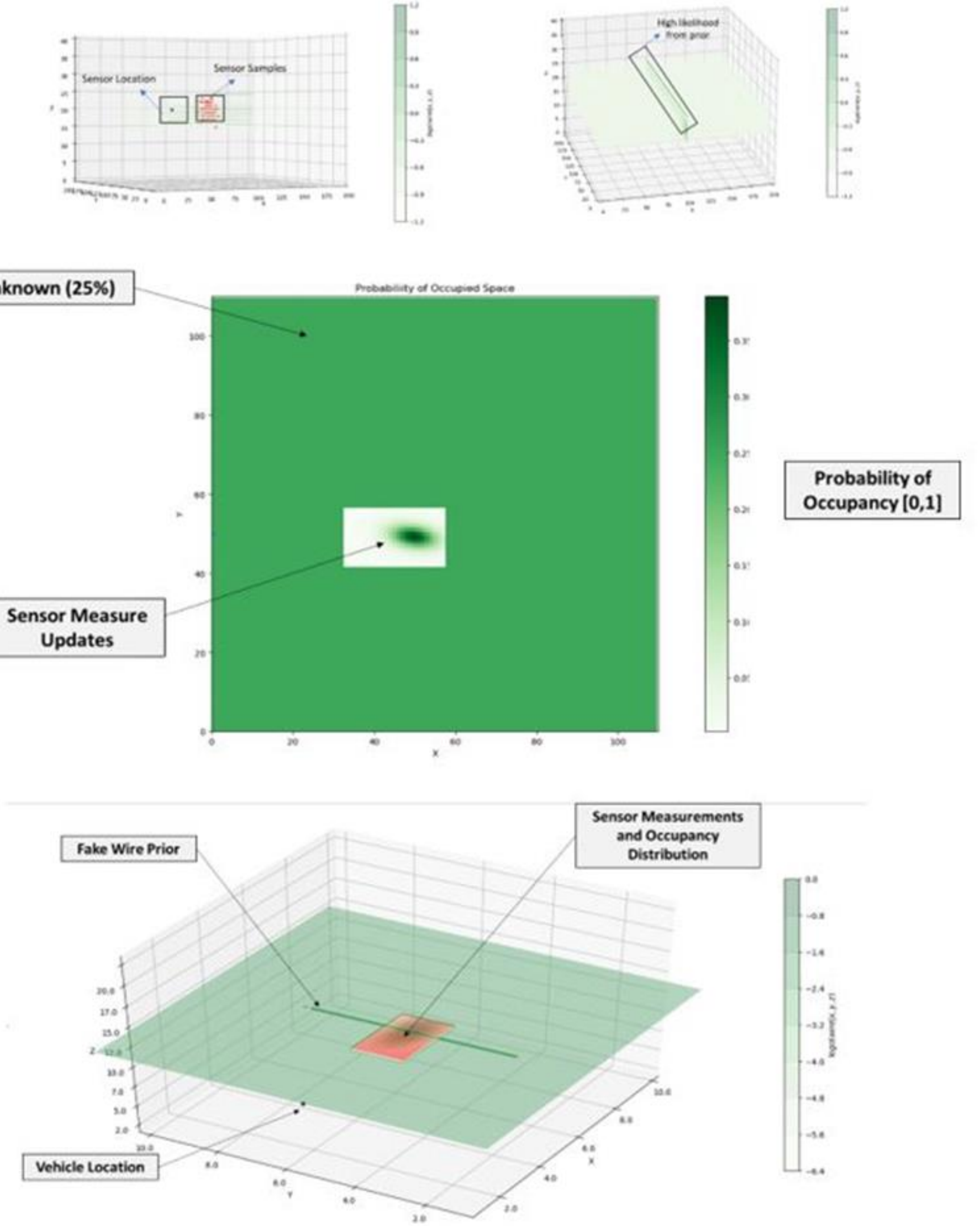


Figure 96. Occupancy grid visualization and measurement fusion

A single experiment is shown in Figure 97 for an idea on the preliminary results. It was assumed that the wire database has a prior confidence of 65% and that measurements had a Gaussian distribution centered around 75% for confidence in a wire detection. For a ¼ meter resolution, updates were seen for a single cell where the wire should be in the wire database. After 10 measurements, the occupancy likelihood increased to over 90%, which with the preliminary results shows to be high enough to indicate a wire. In as little as 3 measurements, a confidence of over 75% can be achieved.

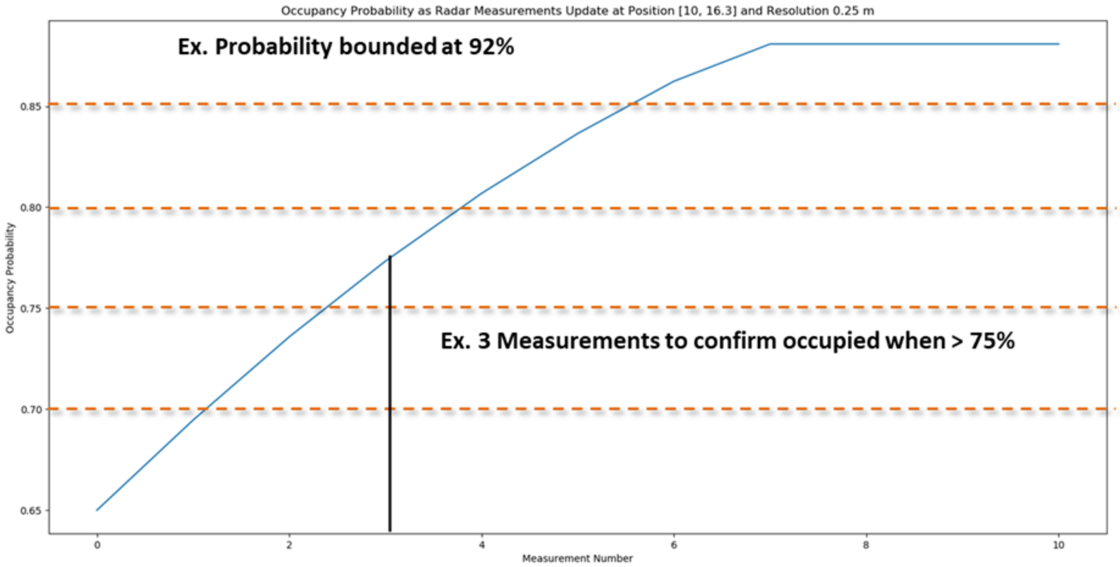


Figure 97. Occupancy grid fusion demonstration plot

The final step is to transform the occupancy grid predictions back to a vectorized dataset, such as a Shapefile (.shp). This step has not been demonstrated yet for a full wire database, since the radar data is limited to a constrained experimental setup, however, the methodology has been demonstrated for a sample portion of the database. A multi-line RANSAC algorithm as shown in Figure 86 is used to form line objects in 3D space from predicted cells of wires that are taken as the center of the occupancy grid cells with a high enough posterior probability of being occupied. The lines are then converted to the same geocoordinate reference system as the initial wire database and saved into a file to be able to display and compare to the true location of the wire.

5.3 Data fusion results and discussion

A series of experiments were prepared to best understand the requirements for the fusion of a wire database and radar data of powerlines. Two phases of results are shown here including parameter optimization of the algorithms and data structures and then sensitivity analysis and performance of the data fusion pipeline over various assumptions.

5.3.1 Algorithm parameter optimization

The data fusion pipeline relies on several algorithms to make conclusions on where a wire may or may not be located as based on the information available. Therefore, it was of interest to this work to investigate how the parameters of the algorithms and inherent data structures change the computational requirements and the prediction performance. In the following sections both the radar detection algorithm parameters and the occupancy grid parameters were investigated.

5.3.1.1 Wire reflectance threshold

The reflectance value is limited by a threshold which controls the tradeoff of noise vs. information. The histogram in Figure 98 shows the μ +/- 3σ of the data's absolute magnitude of the real component of the reflectance. Visualization of the binary threshold at different values is seen in Figure 99, where the top left corner would reflect a high threshold and the bottom right would reflect a low threshold. The inherent question is where to place the threshold so that noise is reduced, but the important information remains.

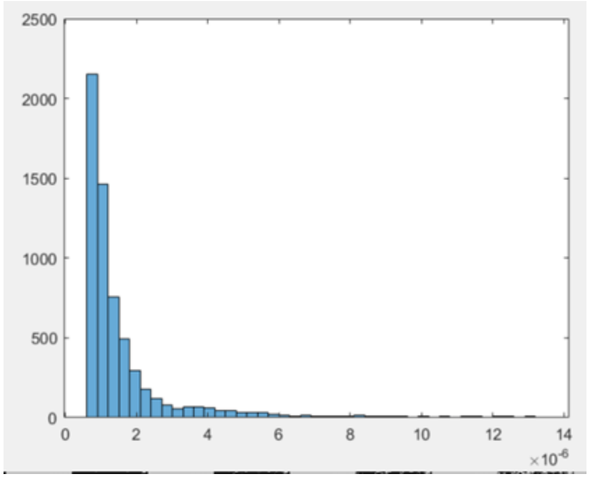


Figure 98. Histogram of reflectance values used for thresholding

A series of tests were run to examine how the threshold changes the number of potential wire pixels that exist in the data. The chart in Figure 99, shown in log-log axes, displays how the number of wire pixels indicated decreases as the reflectance threshold increases. This produces three regions defined as "low noise", "tradeoff", and "high noise." Since experimental data is limited and was collected in a controlled environment without obstructions, the decision was made to operate in the *tradeoff* region to limit the amount of noise in the classification step, but not remove it completely since the experimental data was not yet exemplary of realistic scenarios. The threshold is selected as 2.3e-07.

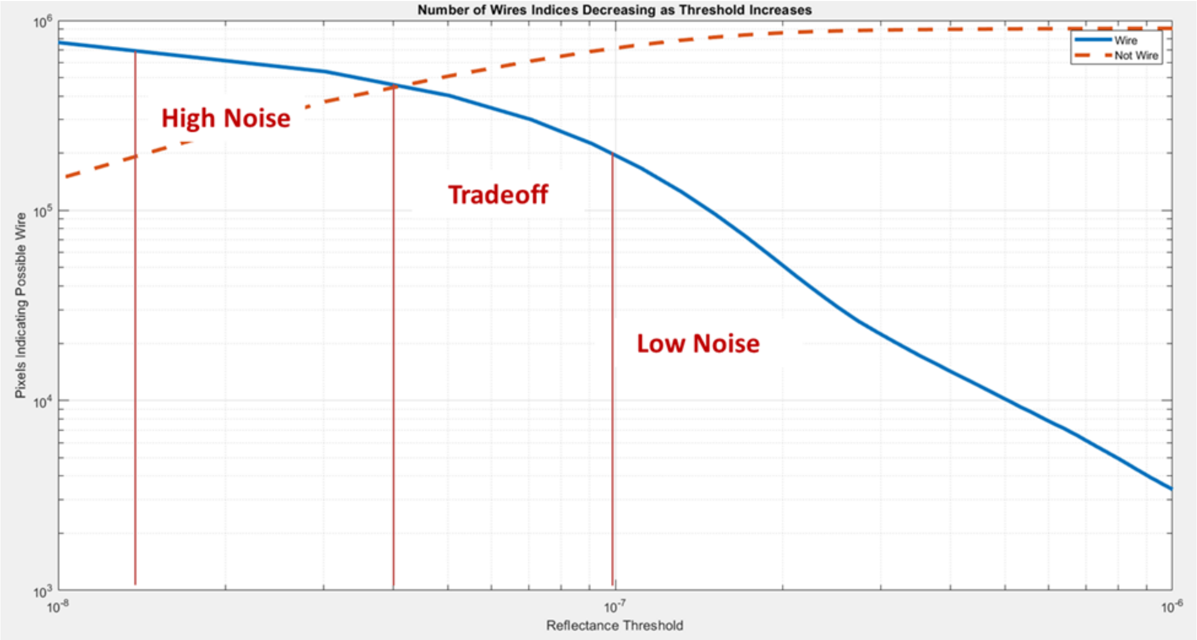


Figure 99. Variation in noise by shifting the binary threshold value

5.3.1.2 Wire detection classifier

The Hough line detection method can successfully detect linear from the binary threshold data, however, knowledge of where the line should be is required to find which of the lines is the wire. In Figure 100, the Hough line transform is shown to produce many potential lines. Therefore, a classifier is selected to differentiate the line that best represents the wire.

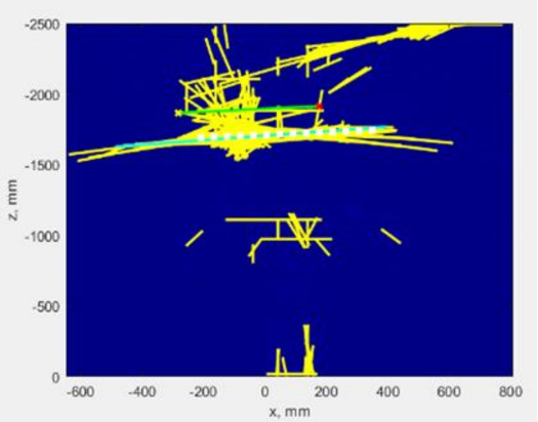


Figure 100. Hough line transform detected linear from radar data

Previous work by Ma et al (Ma, Goshi, Shih, & Sun, 2011) used an SVM classifier and Bragg pattern features to detect wires in radar data. The SVM classifier and 14-dimensional feature vector that represents the Bragg-features is detailed in the background section. That methodology is leveraged here and an investigation of the parameters of the algorithms are conducted.

After compiling the training data from Iowa State University, three datasets were made available. These detections were all from 2 meters away, but include the wire at angles of 0, 5, and 10 degrees. Training involves solving an optimization problem, where we seek the lowest loss (minimization problem). The 338 observations and around 3 seconds of training resulted in a total class loss of 0.075. The Bragg features for 8 of the 14 most important features are shown in Figure 101, with the orange dots classifying a wire and the blue dots classifying no wire (note that the support vectors with missing colors is from a visualization issue). Circles are drawn around the support vectors. Because of the nonlinearity of the classes and the limited training data, there are many support vectors.

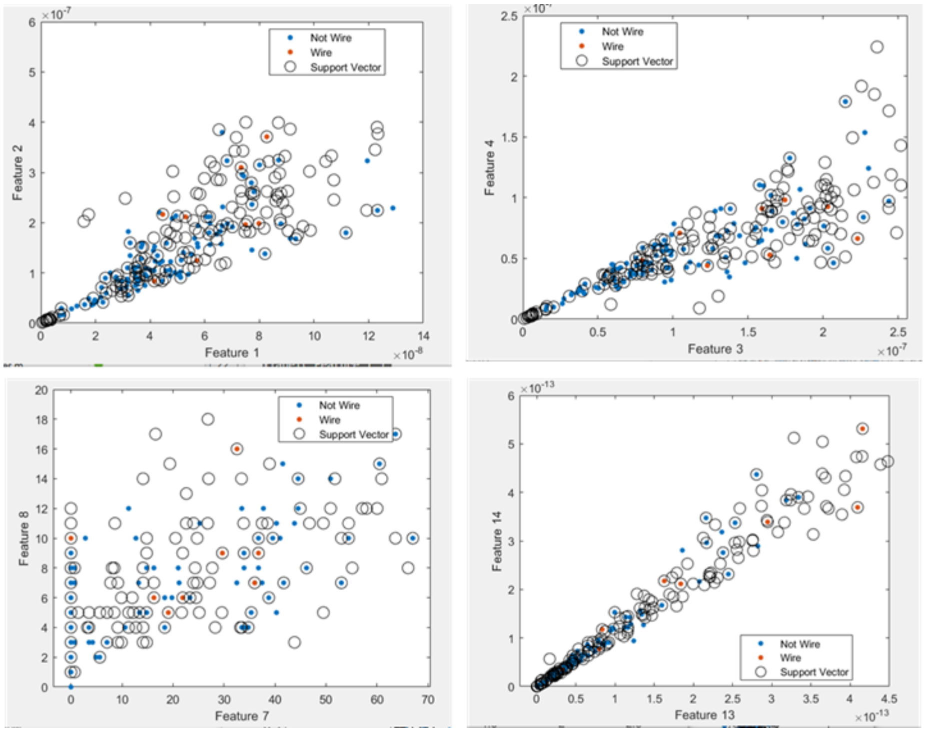


Figure 101. Bragg features from SVM classifier with support vectors and classification

5.3.1.3 Occupancy grid parameters

Computational limitations force constraints on the selection. This comes from time and memory limitations. For example, the grid computations scale polynomial-time with increase in resolution. For this experiment the occupancy grid was randomly selected to be between 50 – 200 meters length and width, and 10 – 30 meters high. Each resolution ran 30 times to sample this distribution accurately. Each run involves initializing the occupancy grid and then updating the probabilities using the prior from the wire database.

As shown in Figure 102 for the case with less than a quarter meter resolution, limitations begin to occur with run times approaching 3-5 minutes and memory exceeding 400 MB. In addition, the benefit in large resolutions reaches a point of diminishing return as it gets above 1 meter. Therefore, the choice is made to stay in the range near 0.4 – 0.6 meters resolution to allow for larger environments or onboard processing to be done in the future. A more detailed examination will be needed in future research when computational limits are more explicitly introduced.

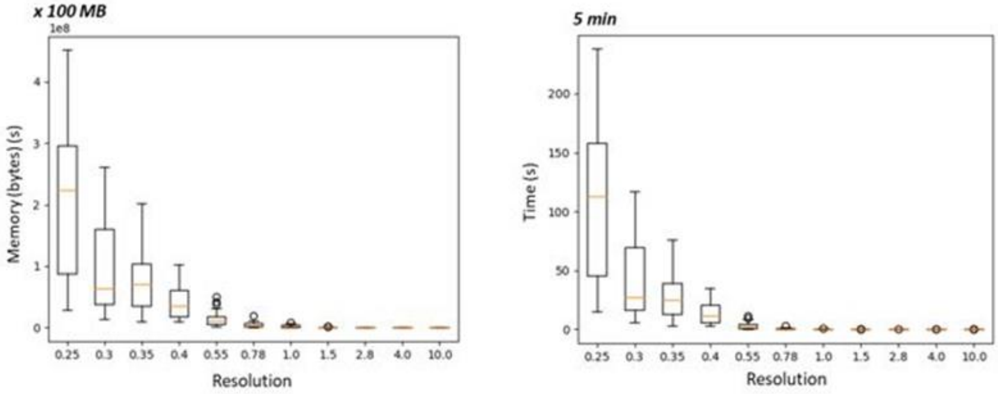


Figure 102. Memory and time ranges for occupancy grid resolution choices

5.3.2 Sensitivity analysis and scenarios

The methodology and software environment provide a foundation for evaluating how an occupancy grid environment changes based on prior information from a wire database and updated information from radar data. Three experiments, detailed in the following sections, were planned for investigating this further and determining the key indicators and minimal requirements for future real-world testing.

The wire database prior is assumed to a uniform distribution with confidence $P'(wire)$ with a buffer radius R_{wd}. The radar measurement data is assumed to be a Gaussian distribution with a max likelihood $P'(wire|radar)_{max}$. Radar observations are assumed to be recursively updated and repeated with the same data for N_s events. See Figure 103 and Table 9 for more details.

Table 9: Data fusion variables of interest

Variable	**Details**	**Range of Interest**
$P'(wire)_{max}$	Prior distribution max value (normalized to 0-1)	(0.50, 1.00)
R_{wd}	Wire database buffer radius	(1,15) meters
$P'(wire|radar)_{max}$	Measurement distribution max value (‘’)	(0.50, 1.00)
N_s	Number of observations of same radar measurement	(1,10)

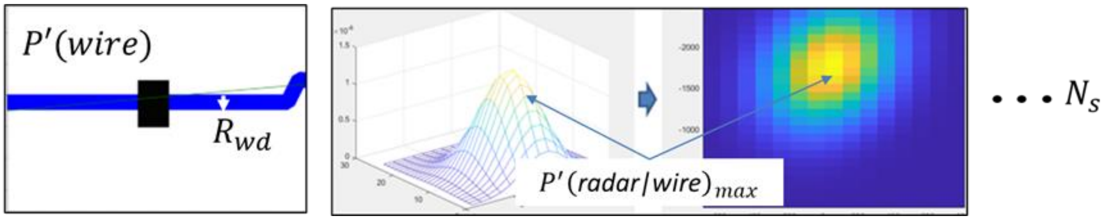


Figure 103. Variables of interest for data fusion experiments

5.3.2.1 Single observation demonstration

A simple set of results to better understand future plots was first demonstrated. A simple case was run for the wire radar measurements max probability, $P'(radar|wire)_{max}$, increasing from 0.5 to 0.95. The resulting posterior distribution, $P(wire|radar)$, increases in confidence as it approaches a bound under 1.00. As expected, a higher radar confidence, or higher likelihood, approaching 0.95 results in larger posterior estimates for a cell that is in the center of the radar measurements distribution, which is where the true location of the wire is, as depicted in Figure 104.

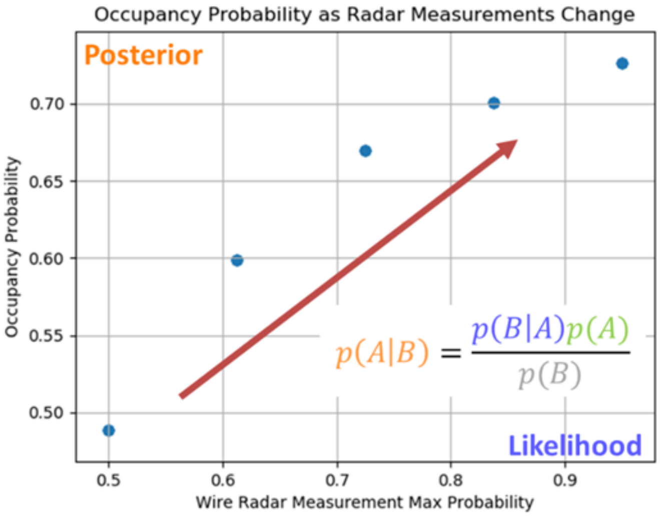


Figure 104. Single observation update demonstration

5.3.2.2 Repeated measurements

The number of measurements and the prior distribution of the occupancy grid are critical to an accurate posterior prediction of wire locations. In addition, the predictions depend on the occupancy grid resolution and the minimum posterior confidence to indicate the wire to a pilot. An example is shown in Figure 105 where two locations, one with and one without the wire, are updated with repeated radar measurements. In real world scenarios, the radar data would be stochastic, but the limited experimental data requires constant measurements for these experiments. The prior distribution is set to 60% and the grid resolution is set to ½ meter. As expected, the location with a wire is indicated from the radar data and increased the occupancy probability from 60% to 95% after 10 measurements. Meanwhile, the location without the wire decreases quickly to less than 10%.

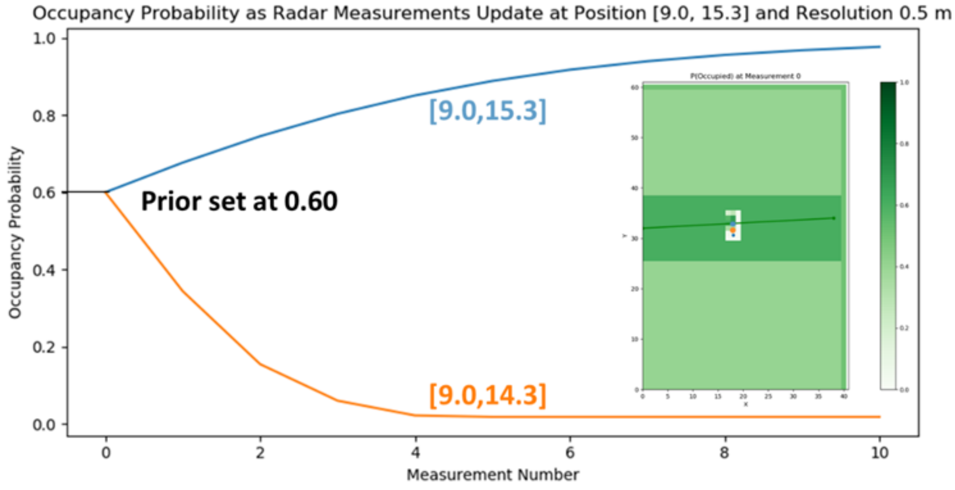


Figure 105. Changing posterior based on recursive radar updates at two locations (blue: wire exists, orange: wire does not exist)

It should be noted that the ability to distinguish the location with and without the wire shown here is dependent on the resolution. For instance, if a larger resolution were used that may converge faster, a larger region of space would indicate the probability of a wire existing there. This will need to be investigated further in future research to determine the appropriate resolution for providing fast and informative warnings to pilots.

5.3.2.3 Wire database prior and radar maximum likelihood

Both the prior distribution and the radar data likelihood have an impact on the posterior distribution, in other words a new prediction on whether a wire exists or not. A simple plot can verify that the better the radar measurement confidence, the higher the posterior probability is. However, the more interesting insight is how these dependent variables change the posterior probability altogether and what are the most important indicators for success. A few conclusions can be made from Figure 106. Sensor confidence and prior confidence combined can guarantee a wire with a single observation. Furthermore, three or more samples always guarantee confidence with the resolution and priors and sensor measurements assumed here.

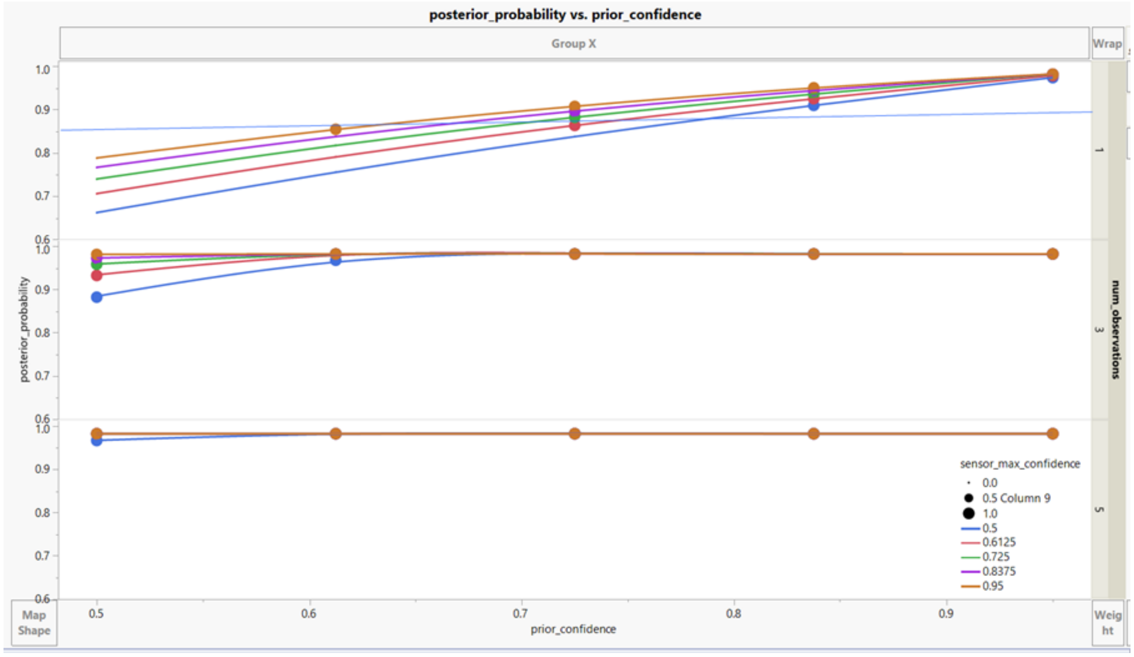


Figure 106. Plot of posterior probability and prior confidence organized by observation count and sensor confidence

Figure 107 shows a plot of posterior probability and prior probability distributed over multiple observations, organized by sensor confidence, and color-coded red if the posterior meets the 0.85 target. Figure 107 shows that increased prior confidence and sensor confidence lead to higher success in achieving the 85% confidence target. Specifically, the 85% confidence target indicates the need for high sensor confidence, and a higher prior can help achieve the confidence quicker.

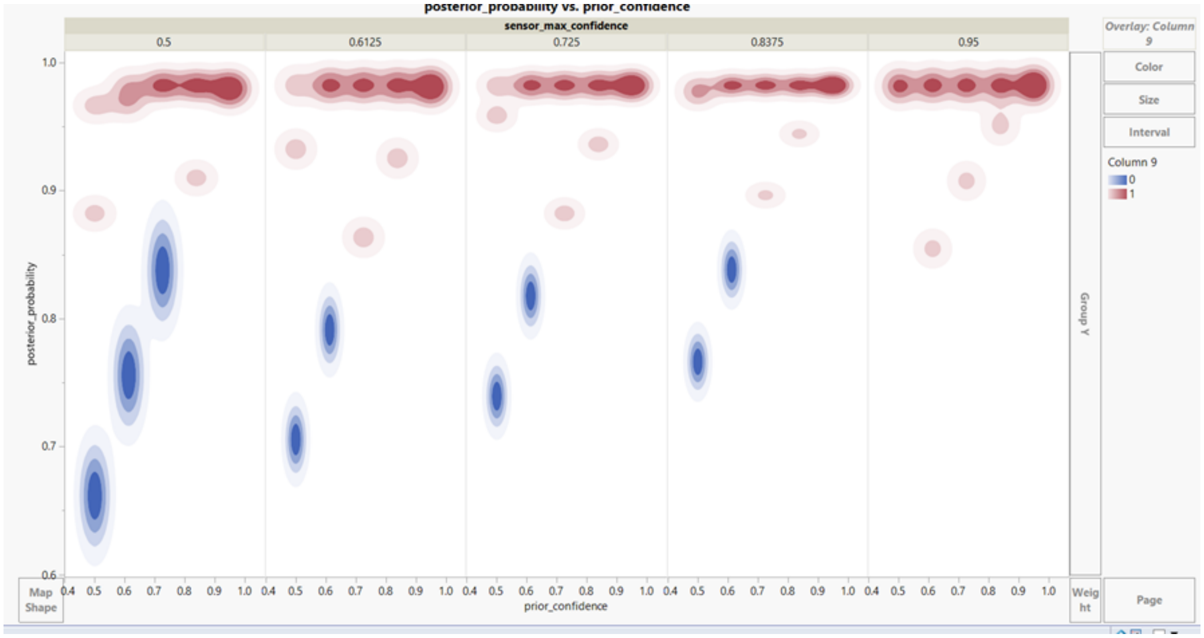


Figure 107. Plot of posterior probability and prior probability

The capabilities demonstrated here can now be investigated further to provide specific metrics required for wire database and radar measurement performance. For instance, to avoid false negatives with 95% confidence target for indicating a wire, the analysis suggests one of three things: a minimum of 3 – 5 observations need to be performed, a sensor confidence of over 80% is required, or the database prior needs to be above 70%. On the other hand, if false positives are of a greater concern and a 70%-confidence metric for indicating a wire is chosen, then the analysis shows that a single measurement is successful with sensor confidence above 60%.

5.3.3 Final demonstration

The final experiment was defined to demonstrate the full pipeline for wire database and radar data fusion and provide insight into how well the data fusion methodology works. The pipeline was detailed and optimized in the previous two sections and is now used to form an updated section of the wire database that can be compared to the truth and prior predicted location of a wire in the local airspace. As explained earlier, the RANSAC algorithm was used to reconstruct a vector, in this case a 2D line, to represent the wire. The accuracy of the line depends on the parameters of the algorithms, in particular the resolution and occupancy threshold level selected.

The new wire prediction was compared to the wire database prediction through five metrics. The error between the new line and the truth line was compared to the error between the wire database line and the truth line. The error was divided into the minimum and maximum error, as well as the mean and standard deviation of the difference in the lines using a discretization at the same resolution as the grid. The performance was evaluated on the percentage improvement in the mean and max errors. In addition, a buffer zone is defined, just as in the wire database, that represents the area of uncertainty around the predicted line. The size of the buffer was compared and evaluated on the percentage improvement after data fusion. The parameters for the final demonstration are shown in Table 10.

Table 10. Final demonstration parameters

P' (wire)	0.65
R_{wd} (m)	7
$P'(\text{wire}\|\text{radar})_{max}$	0.75
N_s	9
Grid Resolution (m)	0.01
Occupied Threshold	0.80
Prior Buffer (m)	7

Figure 108 shows the mean, standard deviation, and max distance metrics changing over observations. While more samples usually mean better occupancy threshold values, under uncertainty this could increase the number of occupied cells and make the RANSAC line formation worse. Therefore, there are two jumps in distance errors at sample 2 and 7. In real-time use, we of course do not know the true location of the wire, and therefore experimental testing could help in tuning the line formation algorithm. Future improvements could investigate the optimal selection of these parameters and investigate alternative methods that may weight the RANSAC line fitting algorithm with the posterior probability values or other informative features.

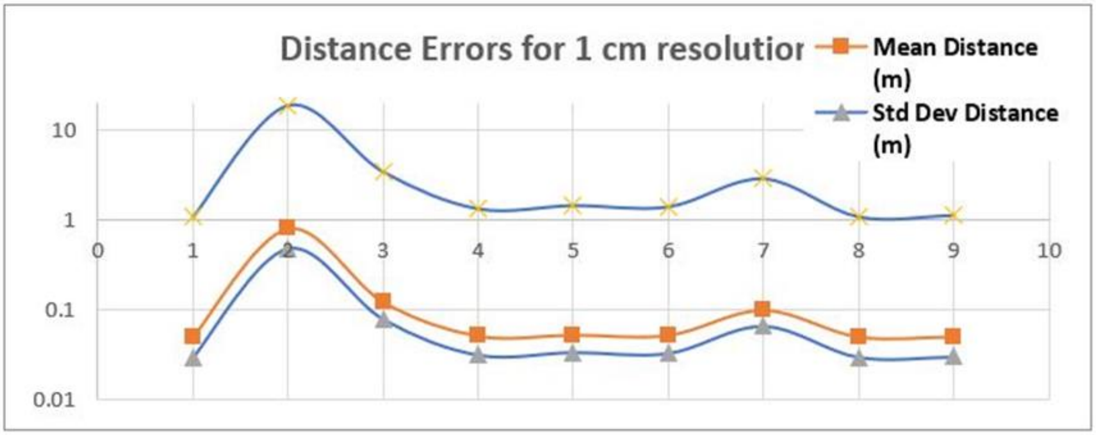


Figure 108. Distance error metrics for final demonstration at 1-centimer resolution

The occupancy grid truth, prior, and predicted line locations are shown in Figure 109. The updated buffer radius is much smaller and still contains the true location of the wire, which is expected from adding sensor data through fusion. Table 11 details the data fusion results, which indicate a 53% decrease in the maximum distance error, a 70% decrease in the mean distance error, and a 94% decrease in the uncertainty bound for a 1-centimere resolution grid with the experiment parameters from Table 10. Therefore, it can be concluded that the data fusion from an onboard radar sensor can improve the position of a wire prediction and decrease the uncertainty.

Table 11. Comparison of wire database and data fusion predicted line location

Metric	**Database**	**Fusion***	**% Change**
Min Distance (m)	0.05	7e-13	-
Max Distance (m)	4.46	2.06	-53%
Mean Distance (m)	2.2	0.67	-70%
Std Dev Distance	0.95	0.47	-50%
Buffer (m)	7.00	0.45	-94%

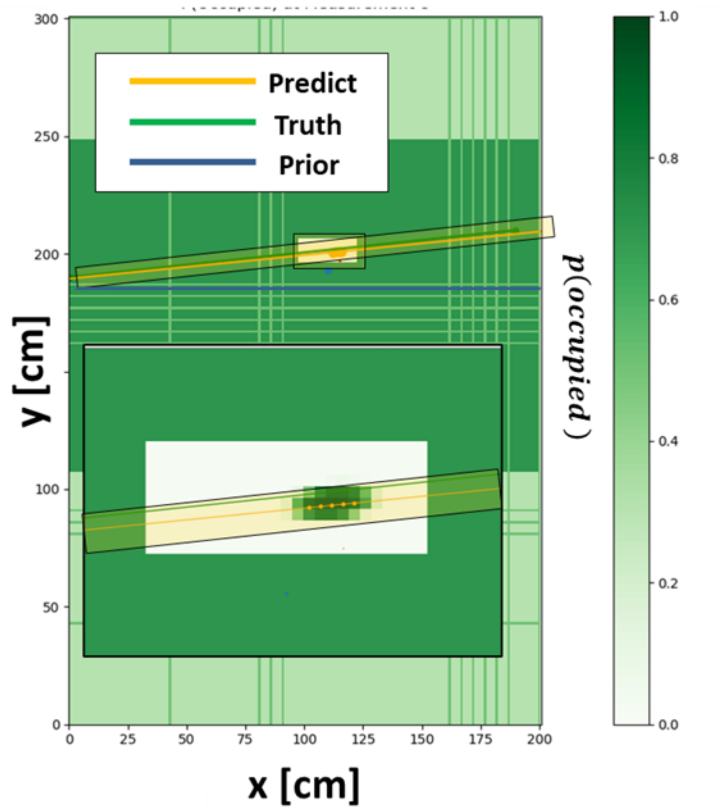


Figure 109. Data fusion line prediction and buffer after final demonstration experiment

5.4 Data fusion conclusions and future work

The data fusion pipeline was developed to investigate how radar data could be used to improve wire databases in real-time or offline. This work approached the goal by developing a simulation environment which leveraged data fusion algorithms to provide insight into the computational and logistical requirements for implementing data fusion in combination with a wire database. A data fusion pipeline was developed that leveraged a Support Vector Machine classifier with Bragg features, an occupancy grid fusion data structure, and a module for matching the wire database with experimental data to demonstrate the approach with real data.

Three key research outcomes were made from the experiments. One, the algorithm parameters are sensitive to the environment and grid size, but in general the target should be to operate in environments without many obstructions and where ½ - 1 meter resolution grids provide enough accuracy to prevent wire strikes. Two, the wire database prior and measurement confidence assumptions are critical to the posterior distribution of wire locations. For example, an 85% posterior confidence target indicates the need for high sensor confidence, while a higher prior can achieve the confidence quicker. In addition, the number of observation targets can be critical in deciding sensor likelihood requirements, with only 3-5 observations needed to guarantee confidence in some scenarios. Three, the full data fusion pipeline can reform sections of the wire database and reduce line prediction error by greater than 50% and the uncertainty buffer by up to 94% when using a 1-centimeter resolution. In addition, the uncertainty is reduced to as little as 1-meter. The wire database can continuously improve to eventually guarantee pilot safety when in use.

Three key conclusions were found from the data fusion pipeline. One, the wire database prior and the radar measurement confidence assumptions are critical to the posterior confidence of whether a wire is in the local map. This links directly to the question of whether to indicate a wire to a pilot. This will need to be investigated for what confidence value correlates to the appropriate false positive/false negative rate and the pilot reaction time. Two, these assumptions result directly from the wire database formation algorithms and data, and the radar detection hardware and software. The fusion tools developed here can be used to find metrics for those two tasks so that improvements can be made that at least meet minimum standards. Three, occupancy grid frameworks are flexible to adapt to the desired resolution/accuracy and the type of measurements available. This means that future advancements could be included for new sensors, other obstructions, and real-time application.

Among the assumptions already detailed, it should be noted here that three key components are not explicitly modeled yet: speed of the vehicle, large distances between sensing data, and obstructions in the sensing data. These will require updated experimental data and eventually data collected from experimental flights. Further improvements could be made to the environment and to reduce the assumptions made. Some of these are discussed here to provide a foundation for future work.

Future methods to investigate include particle filtering for online sensor fusion, data association or map matching methods for fusion to the wire database, Gaussian Processes for handling spatial datasets, and probability hypothesis density filters have shown promise with road maps. The focus thus far has been on offline fusion of a mapping mission, however, future insight will need to consider real-time updates and the fusion of more sensors, for instance cameras. A common algorithm for this is the particle filter. Also, height has not been addressed yet since we only have access to 2D radar data and the wire database assumes a constant height. Therefore, advancements into the database and radar data must be made before height is included. The occupancy grid has already been setup to include the third dimension of data.

Additional radar data could improve the training of the wire detection classifier and enable more experiments to be run. Specifically, indoor and outdoor data collected at a variation of distances and angles, with and without obstructions. Similarly, more wire database locations would promote more insight into where and when data fusion is helpful. In addition, a simulation environment for testing is needed for more experimentation and rapid insight into the requirements and limitations of the current methods. Simulated radar data could prove very useful using packages such as *radarsimpy*[11]. Furthermore, the M&S environment could allow for additional simulated sensors, real-time flight, and even pilot response and visualization.

Next steps should also explicitly address cases with multiple wires and additional clutter in the environment. The current data pipeline assumes a single line object that needs to be detected and transformed into a gaussian distribution. However, all algorithms other than the gaussian distribution formation can already handle multiple targets. Thus, the only change would be to modify the gaussian to be class-dependent or formed from a Gaussian Process or Mixture of Gaussians for instance.

6 Sensor package design

6.1 Introduction

The goal of the sensor package design portion of this project was to outline the design of a frequency-modulated continuous-wave (FM-CW) radar system capable of detecting and locating wires that are potentially in the flight path at a minimum distance of 15 meters, so that a warning can be communicated to the pilot. The specifications of this design include parameters such as: frequency of operation, antenna gain (i.e., antenna amplification), and transceiver detection sensitivity.

In pursuit of this goal, several steps were taken to properly evaluate the efficacy of using a microwave radar system for wire detection. In this work, a literature review was conducted, electromagnetic (EM) simulations, and indoor and outdoor experiments with the available laboratory equipment were conducted. Full-wave 3D electromagnetic simulations were first performed to evaluate the effect of frequency, polarization, and wire geometry on the detectability of the wires. These simulations were used to obtain a reasonable estimate of the optimal (or near optimal) system parameters.

Measurements, using actual wires were then performed in an indoor setting, using laboratory equipment such a vector network analyzer (VNA), in order to experimentally determine the possible system parameters. Many experiments were performed on some actual wire samples

---

[11] https://github.com/rookiepeng/radarsimpy

using different operating frequency ranges, antennas, wires, and distances between the radar and the wire under test. Following these experiments, limited outdoor measurements of real power lines were performed as well.

In addition to performing these simulations and measurements, previously published works related to power line detection using radar technology were also examined (Sarabandi & Monsoo, 1994; 1999; 2000). This led to some important findings related to the efficacy of using such a radar system. The most critical finding was that the primary mechanism for scattering from a power line is specular reflection. As a result, the ability of a radar system to detect a power line or other long wires (i.e., a guy wire) is highly dependent on the angle between the wire and incident radar signal. Note that specular reflection occurs with smooth surfaces (such as solid wires) and causes radar signals to reflect off the surface away from the radar at the same angle at which the signal was incident upon the surface. Diffuse scattering, in contrast, is caused by rough surfaces and causes signals to scatter in all directions (including back to the radar).

To better explain this issue, Figure 110 shows a simple diagram of the geometry. In order to detect scattering from any given point along the wire, the angle θ between the wire normal and the incident signal must be very close to zero, due to the fact that the specular reflection is dominant in this situation (i.e., there is little diffuse scattering). When the angle between the radar antenna and a point on the wire is relatively large, the wire becomes undetectable. Note that although the amount of diffuse scattering is larger for stranded wires, it is still sufficiently low to inhibit detection. The specific values for the incidence angle are explored later through measurements.

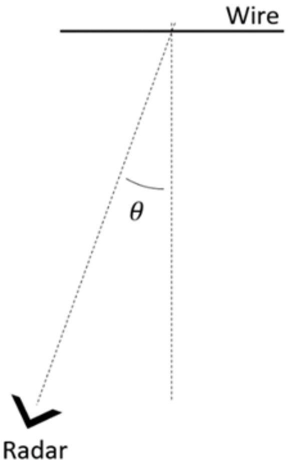


Figure 110. Diagram of an antenna pointing at a wire at an incidence angle of θ

6.2 Electromagnetic simulations

In this section, a set of limited results from several relevant EM simulations are discussed. These simulations were all performed using CST Microwave Studio®, which is a full-wave electromagnetic simulation tool. The goal of these simulations was to determine several of the critical optimal radar system parameters, including the frequency range, antenna size, and transceiver detection sensitivity.

6.2.1 Effect of signal polarization and frequency

One desirable feature of the radar system would be the ability to determine the relative orientation of a detected wire, since this information may allow the type of wire (e.g., power line, guy wire, etc.) to be identified. Through these EM simulations, the idea of using polarimetric radar techniques for this purpose was explored (Sarabandi & Monsoo, 1994; 1999; 2000). The basic idea behind these techniques is to consider the polarization of the signal or signals used by the radar to determine wire direction once it has been successfully detected. This method makes use of the fact that long wires can be highly polarizing targets to the incident radar signal (i.e., the wires only scatter when the polarization of the signal is parallel to its preferred orientation or direction). Thus, if the polarization orientation that maximally interacts with the wire is determined, then the wire orientation can be ascertained.

In pursuit of this goal, EM simulations were performed to compute the radar cross-section (RCS) of long wires. By simulating the RCS, which is a measure of the scattered power received by the radar as a relative to the power level of the signal incident on a certain length of wire, it is then possible to compare the detectability of wires when using polarization parallel vs. perpendicular to the wire and as a function of the radar frequency. Figure 111 shows the RCS of a 12.5 mm-diameter wire vs. frequency for two different radar polarizations, namely: one parallel to the wire, and one perpendicular to it. Radar cross-section (RCS) is a parameter that is proportional to the power level scattered towards the radar (i.e., detected) by a target (e.g., a wire). A higher RCS level means a wire is easier to detect. These types of simulations can aid in determining the optimal frequency range to use, depending on the desired capabilities. For example, in Figure 111, at frequencies where the RCS differs greatly between the two polarizations (e.g., at 1 GHz), wire orientation may be determined using polarimetric radar techniques. In contrast, at 10 GHz, the RCS is nearly the same for either polarization, and thus the polarization used does not have any impact on detection diversity.

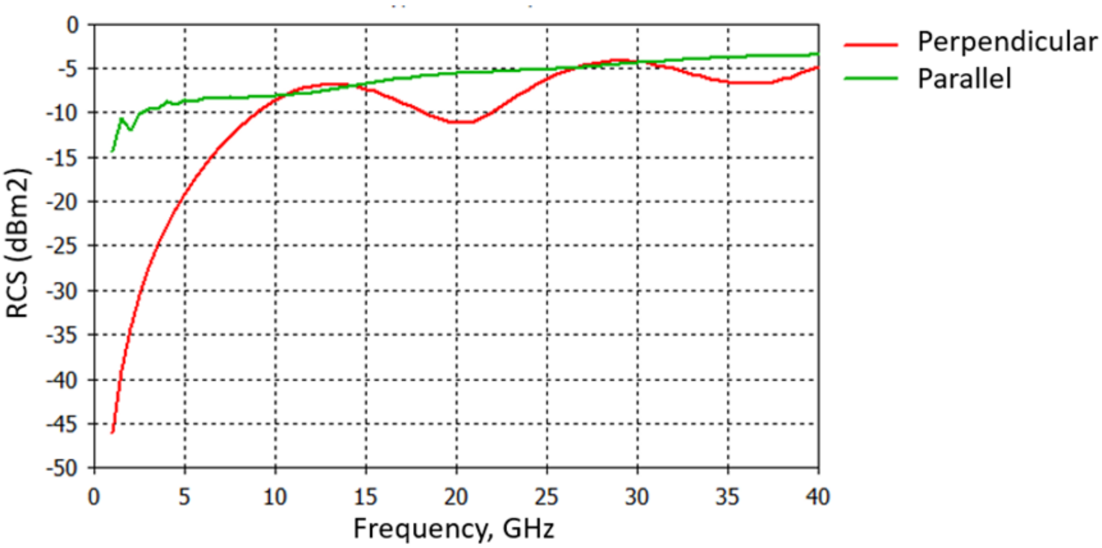


Figure 111. Plot of RCS of a 12.5 mm-diameter wire as a function of frequency using perpendicular vs. parallel polarization

Note that changing the diameter of the wire also impacts the results. Specifically, the ratio of the wire diameter to the wavelength of the signal is the parameter of interest. Thus, a larger (i.e., thicker) wire diameter means a lower frequency can be used to obtain the same results (frequency and wavelength are inversely proportional). Additionally, the geometry of the stranding on the wires has an effect. However, this effect is small as mentioned earlier.

It is worth noting that as the frequency increases, the RCS generally increases before leveling off to approach a constant value. This implies that increasing the frequency should increase the detection capabilities of the radar. Additionally, note that in Figure 111, at frequencies higher than ~10 GHz, the RCS values for parallel and perpendicular polarization approach each other. The consequences of these two observations are that the operating frequencies used should be above at least 10 GHz, and can approach 40 GHz and above. At these frequencies however, using a polarimetric radar system does not allow determination of wire orientation, and thus there is no point in using polarization diversity. Of course, while frequencies higher than 40 GHz may allow for better detection sensitivity, system complexity increases, so this tradeoff must be considered. These guidelines were used to determine the frequency ranges to use in the subsequent experiment, described in the next section.

6.3 Indoor measurements

Based on the conclusions found from the EM simulations, measurements of wires were performed in an indoor setting, using the available laboratory equipment. A vector network analyzer (VNA) is used to perform the function of an FM-CW radar system. These experiments were used to determine the possible radar system parameters. In the EM simulations, we determined that the frequencies used should be greater than 10 GHz, with larger frequencies being better. To this end, additional measurements using a commercially available radar system operating at 70 GHz was used to perform similar measurements. However, these measurements were inconclusive. This was believed to be due to issues related to the provided interfacing software package. Thus, the maximum frequency used for the results shown in this report is 40 GHz, which is the highest frequency supported by our VNAs at the time of measurements.

The VNA generates a signal that is then radiated by the antenna, as shown in a simple diagram in Figure 112. The VNA then measures the backscattered signal (magnitude and phase) detected by the antenna, resulting from scattering by objects in the path of the antenna beam, such as wires (Richards, Scheer, & Holm, 2010). These measurements can additionally be performed over a wide range of frequencies, allowing the distance to any detected scatterers to be determined as well as the amount of scattering (Richards, Scheer, & Holm, 2010). The choice of antenna is also important due to several factors such as the antenna gain and beamwidth.

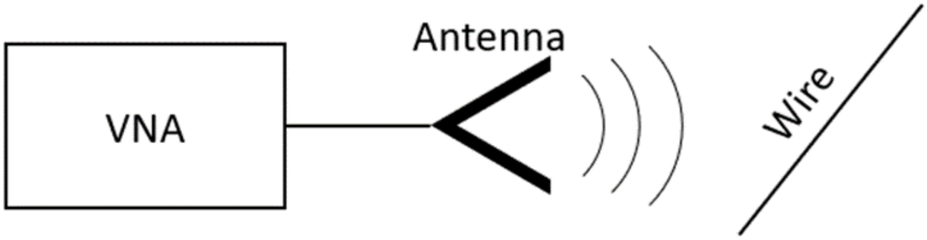


Figure 112. Simple schematic diagram for detecting a wire indoors

6.3.1 Initial experiments

One possible way to implement the overall measurement system is with rotational scanning. This involves using a narrow beam (i.e., a relatively high gain) antenna and physically steering its beam by rotating it up to 360 degrees in increments, while making measurements at each increment, effectively enabling the determination of whether a wire is present in any given direction. Additionally, using wideband measurements, the distance to any detected wire is determined as well, resulting in a 2D image of the surrounding area, where one axis of this image corresponds to the rotation angle, and the other corresponds to the target range (Richards, Scheer, & Holm, 2010). This 2D image can be processed using data fusion algorithms, as described in Section 5. Alternatively, a phased array can be used to electronically steer the antenna beam through the desired range of angles.

In this section, we attempt to emulate the physical antenna rotation through the use of synthetic aperture radar (SAR) techniques (Case, Ghasr, & Zoughi, 2011). A wire was placed on an automated scanning table. An antenna, connected to a VNA, was then raster-scanned over a 2D area at some fixed distance above the wire, as depicted in Figure 113. The data collected from this scanning procedure was then processed using a SAR algorithm to create an image of the wire (Case, Ghasr, & Zoughi, 2011). This has a similar effect to steering the beam of an array of antennas to cover the area containing the wire.

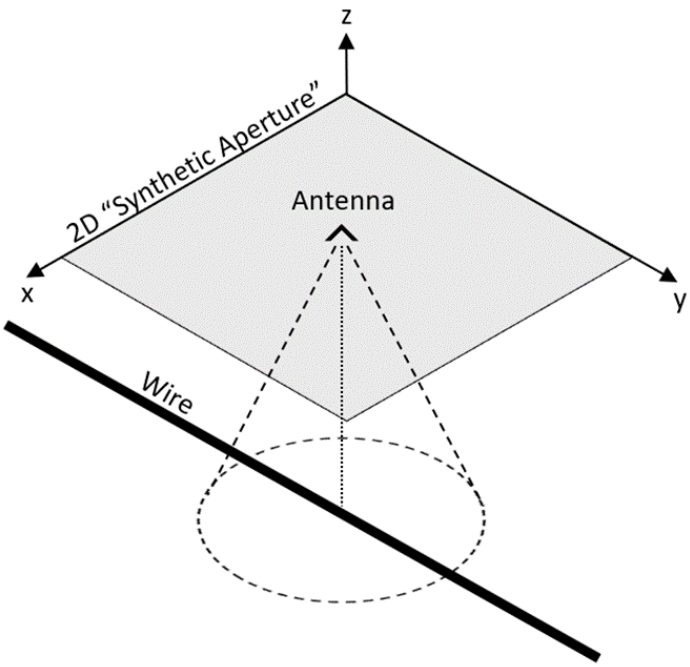


Figure 113. Diagram of an antenna being scanned in a 2D area above a long wire

Two different types of wires were scanned in these experiments. Figure 114 shows a picture of the two wires used. Both wires have a diameter of ~12.5 mm, and one is stranded and one is solid. The diameter was chosen to be relatively close to that of typical wires used in power lines.

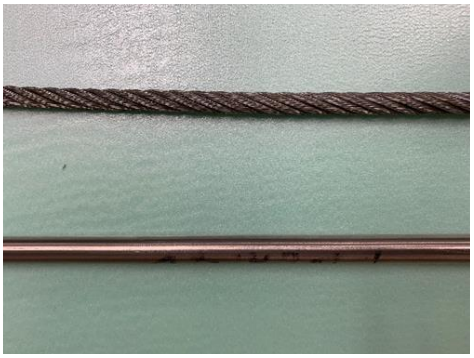


Figure 114. Picture of a stranded (top) and solid wire (bottom) both with a diameter of ~12.5 mm

Many scans were performed to evaluate the effectiveness of this approach. Although several different frequency bands were tested, ranging from 12.4 GHz up to 40 GHz, the best measurement results were achieved when using the K-band (18-26.5 GHz) and Ka-band (26.5-40 GHz) frequency range, where 40 GHz is the highest frequency supported by the VNA in our laboratory at the time these measurements were made. Since the results using the K-band and Ka-band frequency ranges are very similar, only results at Ka-band are shown here. It should be noted that frequencies higher than 40 GHz are expected to work as well. However, as mentioned earlier, results using the 70 GHz radar system were inconclusive, potentially due to software issues.

Figure 115 shows SAR images made of the stranded wire when using the Ka-band (26.5-40 GHz) frequency range and at a distance (between the wire and the horn antenna) of ~250 mm. Parallel polarization is seen on the left and perpendicular polarization on the right to the wire. The white rectangle shows the wire length and placement. These images show the relative scattered signal level originating from various locations in the scene. Note that since the wire diameter is in the order of the operating wavelength (e.g., the wavelength at 30 GHz is 10 mm, and the wire diameter is ~12.5 mm), the relative measured signal magnitudes are quite similar for the parallel and perpendicular polarizations, as expected based on the simulations. The results for the solid wire were also very similar and thus are not shown here.

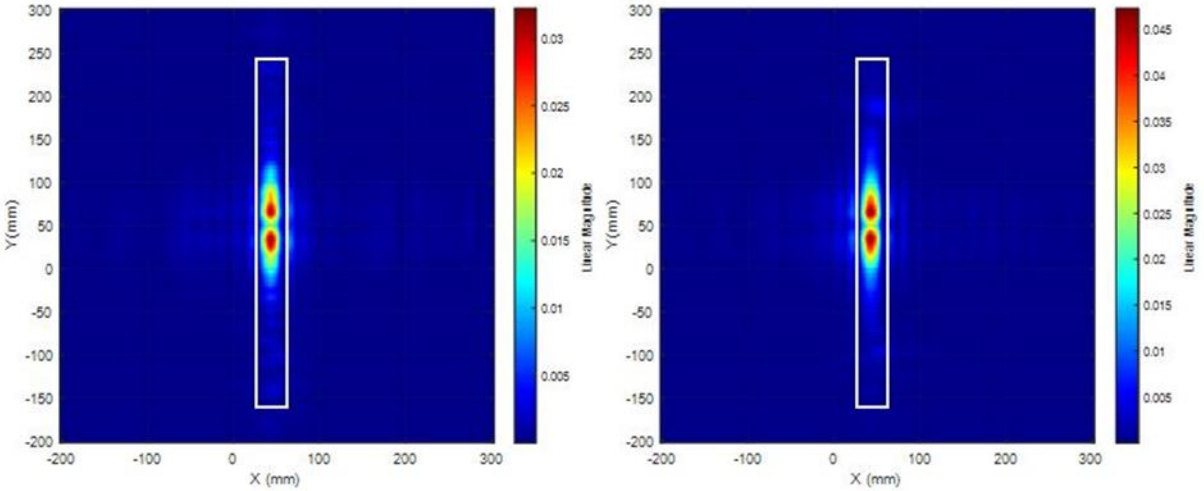


Figure 115. Ka-band (26.5-40 GHz) SAR images of stranded wire using parallel polarization (left) and perpendicular polarization (right)

Note that the wire used is many times longer than the length of the indication shown in Figure 115. From the images, it would appear that the wire length is ~100 mm, when the actual length is more than 400 mm. However, due to the specular scattering effect mentioned earlier, the scattering from the wire is only detected when the incident signal propagates in a direction perpendicular to the wire length (i.e., specular reflection). This effect is examined more closely in the next section. Additionally, the shape of the indication, which contains two distinct peaks, is also an indirect result of the specular scattering effect. This was confirmed by running additional numerical simulations of this exact scenario.

6.3.2 Effect of incidence angle

As mentioned earlier, the primary scattering mechanism of long wires is specular reflection. Consequently, the incidence angle of the transmitted signal relative to the wire is a critical parameter. In order to measure sensitivity to this parameter, several experiments were performed. Figure 116 shows a picture of a rotary scanning system that was used to perform these experiments. The wire is placed on the rotational stage, which can be automatically rotated to any desired angle. A 1D linear stage was used to scan an antenna, allowing a SAR image to be created. The length and step size of the 1D scan were chosen to be 160 mm and 2 mm, respectively. These values were chosen to provide sufficient resolution and sampling to avoid aliasing issues and other imaging artifacts (Case, Ghasr, & Zoughi, 2011).

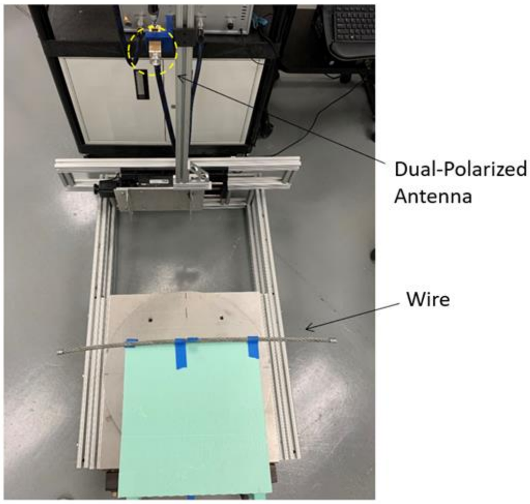


Figure 116. Picture of the setup used to image wires as a function of wire orientation

The setup shown in Figure 116 was used to scan wires using many different orientations. Figure 117 shows Ka-band (26.5-40 GHz) SAR images produced from the stranded wire at two different orientations from a distance of ~575 mm. The yellow line shows the antenna scan path and the white rectangle shows the wire length and placement. Note that the indications in the image do not cover the entirety of the length of the wire, due the aforementioned specular reflection issue. Also, the polarization used was parallel to the wire. The results for the other polarization and for the solid wire are very similar and thus are not shown here. The primary takeaway from these images is that while the wires can clearly be detected, the portion of the wire that is detectable depends on the position of the antenna relative to the wire. More specifically, a particular portion of the wire can only be detected if the line from the radar to that portion of the wire is perpendicular to the wire (i.e., if θ as depicted in Figure 110 is small enough).

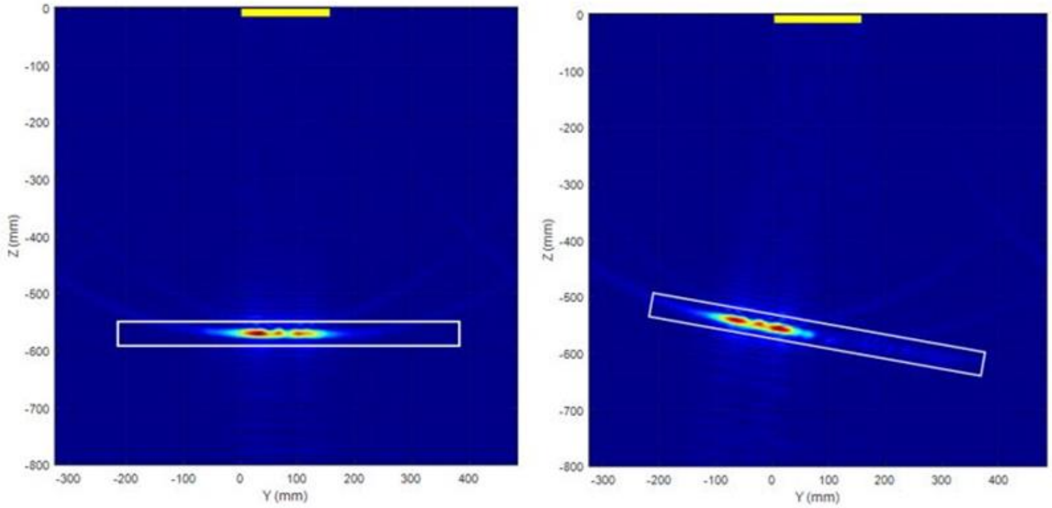


Figure 117. SAR images of a wire at Ka-band (26.5-40 GHz) from a distance of ~575 mm

The results can be summarized as a function of the wire angle. Figure 118 shows a plot of the wire reflectivity extracted from these SAR images as a function of the wire orientation (i.e., θ as defined in Figure 110). These results show that θ must be smaller than about 2-3 degrees in order for the radar to successfully detect the wire. Note that the response for the stranded wire has a slightly narrower main lobe but has side lobes with a higher magnitude. This behavior is expected based on the literature (Sarabandi & Monsoo, 1994; 1999).

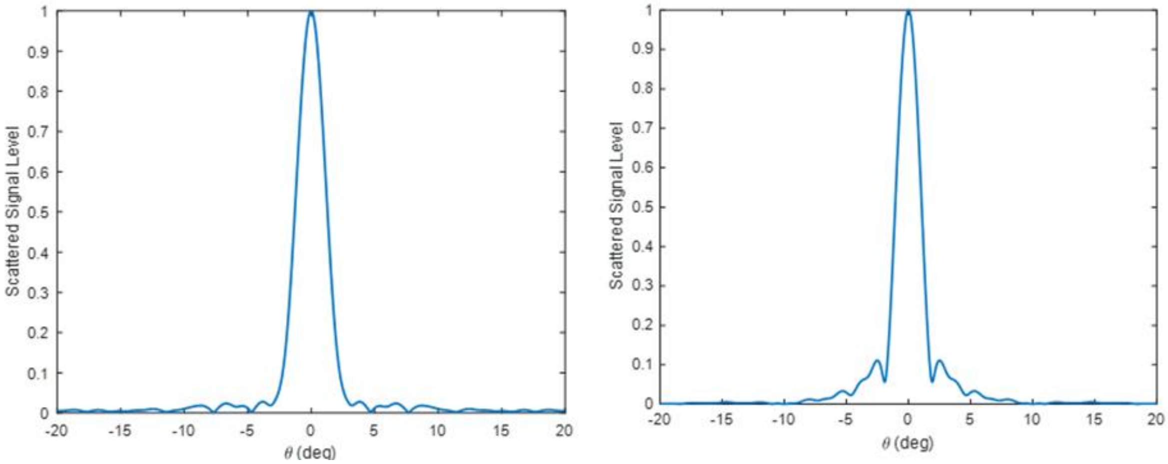


Figure 118. Plot of measured wire reflectivity vs. wire orientation for solid (left) and stranded (right) wires

The imaging procedure was also implemented at several different distances between the antenna and the wires and at the two orthogonal polarizations.

Figure 119 shows a SAR image produced using the same setup at Ka-band (26.5-40 GHz) but at a target distance of 1700 mm. The yellow line shows the antenna scan path, and the white rectangle shows the wire length and placement. Even at this distance, the wire is still detectable.

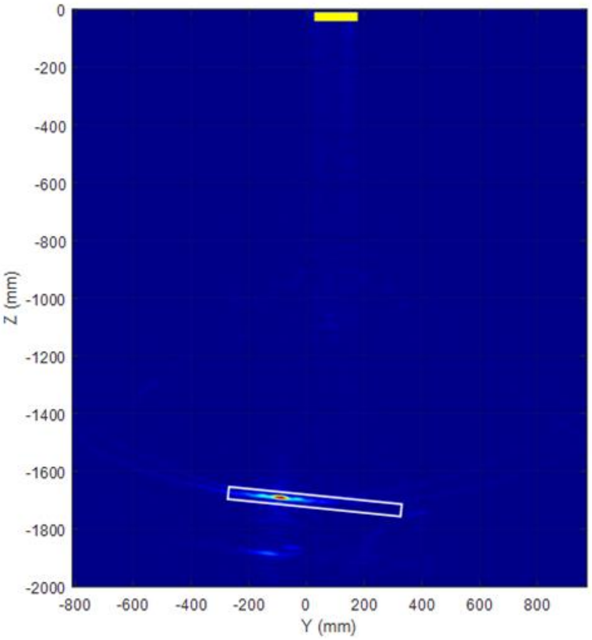


Figure 119. SAR image made of wire at Ka-band (26.5-40 GHz) from distance of ~1700 mm

The measurements shown in this section demonstrated the efficacy of using a radar system operating at Ka-band (26.5-40 GHz) to detect and localize wires. Additionally, the relevance of the wire specular scattering issue was effectively demonstrated. Due to limited laboratory space, measurement could only be performed on the scale of around 1700 mm. Thus, outdoor measurements are shown in the next section that are made from a much greater distance.

6.4 Outdoor measurements

Due to the limited radar to wire distances that could be experimentally measured indoors and in order to make measurements of real power lines, the experiments performed in the previous section were repeated outdoors, where larger distances between the radar and the power lines can be realized. The procedure for these experiments is the same as in the previous section, except that they are performed outdoors and on real power lines that are much further away (more than 15 meters).

6.4.1 Power line measurements using 1D linear scans

Figure 120 shows a top-down view (satellite picture) of the experimental setup. In this picture, the relative location of an existing power line (with three power conductors and a fourth serving as lightning arrest line) can be seen relative to the measurement (radar) system. The scanning system was placed at a ground-range of about 20 meters from the power lines. Note that the power lines have a different elevation than the radar system, and thus the distance (i.e., range) between the radar and the wire is larger than 20 meters. For these experiments, a VNA connected to a K-band (18-26.5 GHz) antenna was attached to a 1D linear scanner. Note that K-band was selected over Ka-band due to the fact that our highest gain antenna operates only in K-band. From the scan data, images were created, as explained in the previous section. Note that since the targets were much farther away than in the indoor experiments, a larger horn antenna was used to provide higher gain (~25 dB gain), as shown in Figure 121.

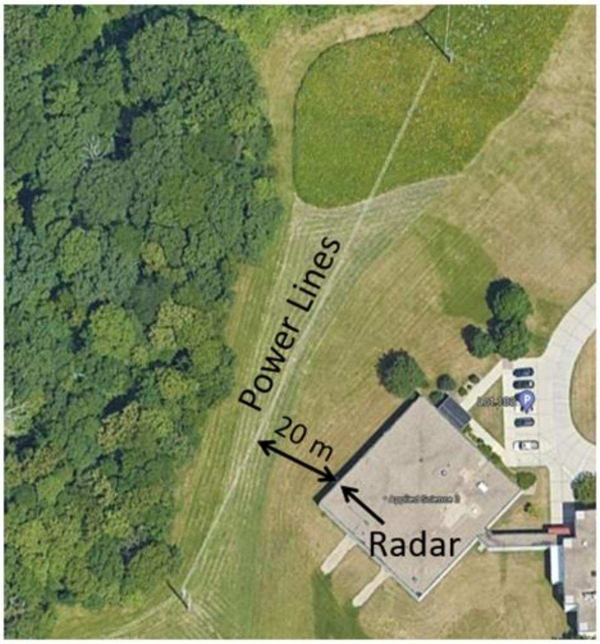


Figure 120. Top-down view of power line detection experimental setup

However, before showing results of the power lines, results of some initial experiments are shown to verify the methodology used here. Figure 121 (left) shows a picture of the scan setup being used to image a highly reflective target placed on a pole ~10 meters away from the radar system. Figure 121 (right) shows the resulting SAR image, where the target can be clearly identified at the correct location. This experiment helped validate the methodology used for the power line measurements. Note that the horizontal width of the indication corresponding to the target is due to the resolution of the imaging method, which itself is a function of the 1D scan dimension and the range to the target.

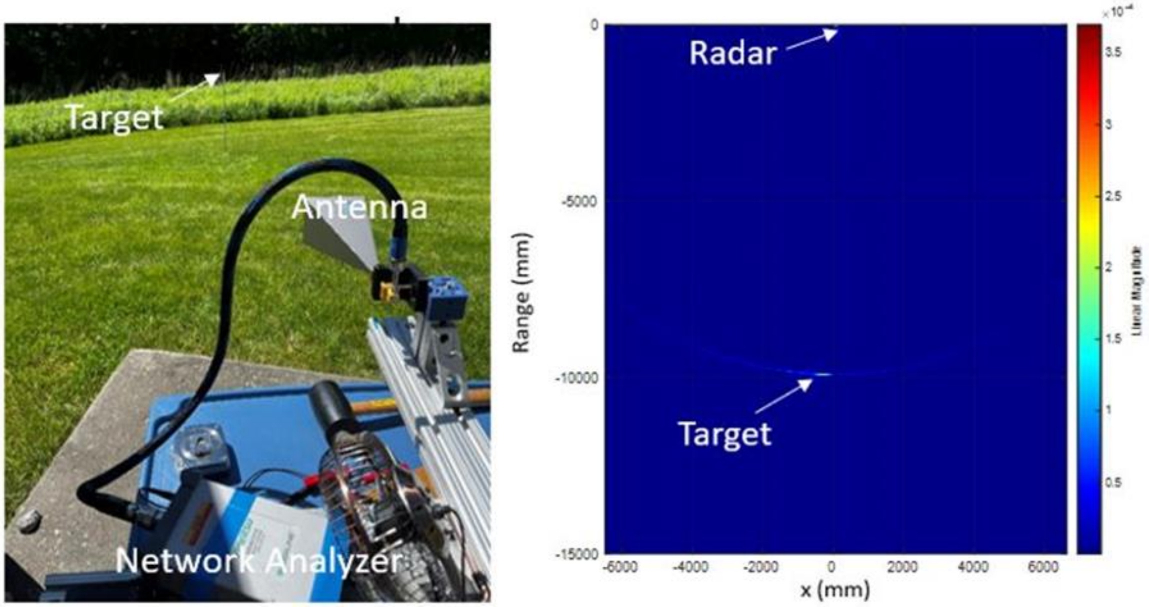


Figure 121. Picture of experimental setup (left) SAR image of target (right)

Next, the same setup was used to image the power lines shown in Figure 120. Note that there are 4 power lines, each with different elevations (i.e., ranges) with respect to the transmitting antenna. Given that there are 4 different elevations, the distance to the radar system is different for each of the 4 lines. Figure 122 shows the SAR image of the power lines. Note that the image colormap is a log scale (as opposed to a linear scale) to better show the relatively low-magnitude indications corresponding to the wires. Additionally, note that the 4th power line is not seen due to the fact that it is outside the beam (i.e., field-of-view) of the antenna.

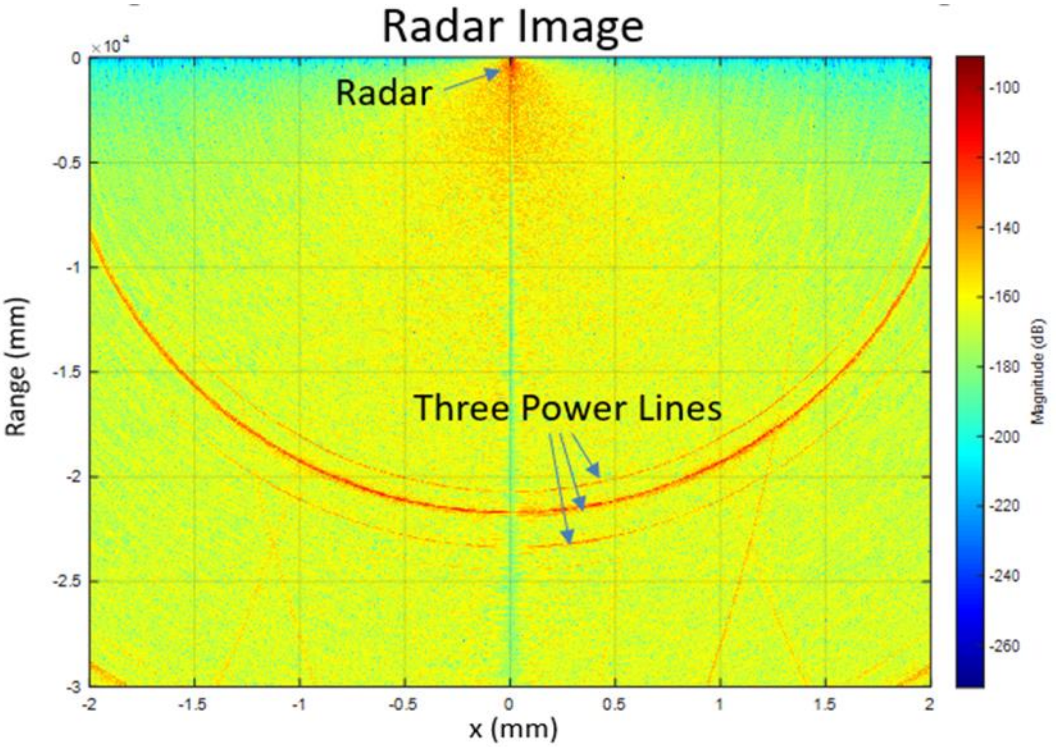


Figure 122. SAR image of the 4 power lines

It is worth noting that the power line indications seen in Figure 122 are in the shape of a circular arc. This is contrary to what is expected since the wires are nearly straight. However, it turns out this effect results from the fact that the wires are specular scatterers, as mentioned earlier, and this effect can be reproduced with numerical simulations. As will be shown in the next section, when scanning the antenna rotationally, this effect is minimized.

6.4.2 Powerline measurements using rotational scans

In this section, the results of using a rotational scanning methodology are discussed. This involved using a high-gain antenna (and narrow beam) that is rotationally scanned to cover the area of interest. To experimentally verify this proposed method, scans were performed using a rotary scanning system. The same setup described in the previous section was used here, except that the antenna was physically rotated (i.e., pointed in different directions), instead of being scanned linearly. Figure 123 shows an image created from the scan data with a colormap in a linear scale. Note that the horizontal axis corresponds to antenna rotation angle over the course of the scan. It is worth pointing out that the horizontal width of the indication is primarily determined by the beamwidth of the antenna used, which is ~10 degrees.

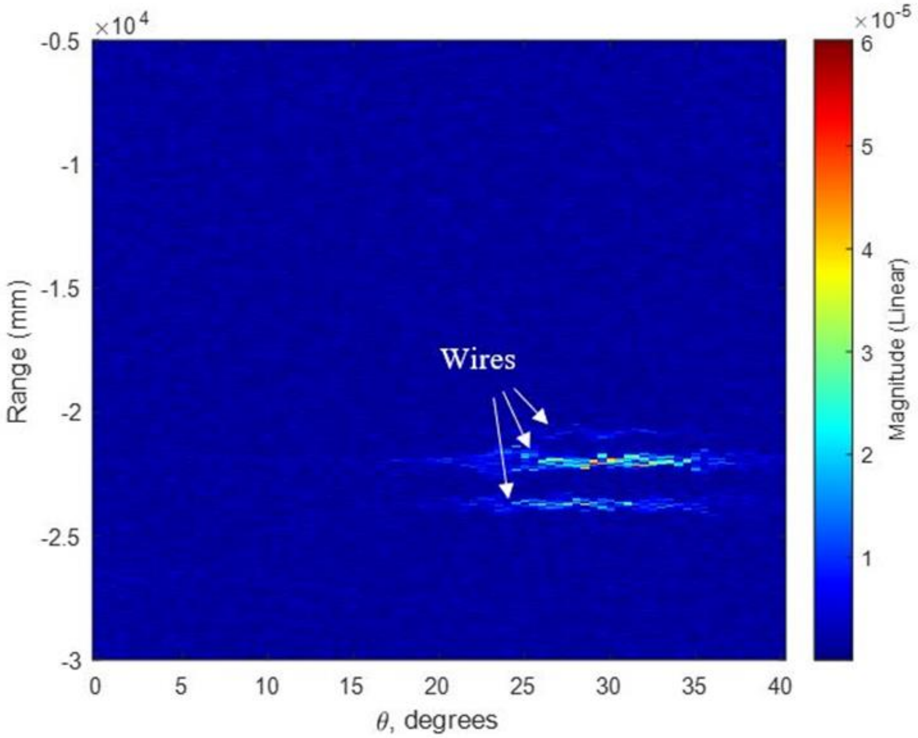


Figure 123. Image of the 4 power lines vs. range and antenna orientation

Since the setup used in this experiment is very similar to the proposed setup, the results of experiments like this one can be used to directly determine required system parameters such as antenna gain and radar dynamic range.

6.5 Proposed radar system design

The proposed imaging methodology involves physically rotating an antenna, connected to a FM-CW radar (transceiver) as depicted in Figure 124, to cover the area of interest. For each antenna rotation increment, the transceiver can perform a measurement to detect any scatterers within the antenna beam and within a range of interest, allowing wires to be detected and localized and as a function of both distance and heading. Note that the antenna beamwidth determines what rotational increment is required, and antenna beamwidth is typically small for high gain antennas. The critical parameters of the radar system include the distance to the wire (R_w), the radar operating frequency range, the antenna gain (G_t), and the transceiver sensitivity (i.e., the minimum detectable reflection coefficient Γ_{min}). Possible values for these parameters can be determined using the measurement results in the previous section.

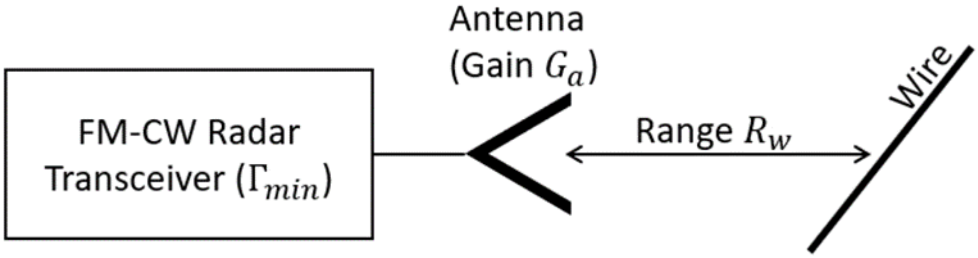


Figure 124. Basic schematic diagram of the FM-CW radar transceiver and wire

Based on the experimental results, the magnitude of the measured reflection coefficient (i.e., the magnitude ratio of the backscattered signal to the transmitted signal) resulting from a power line at a distance of ~21 meters and using a 25 dB K-band (18-26.5 GHz) horn antenna was about $|\Gamma| = 16\,x\,10^{-5}$. From this result, the required antenna gain vs. the transceiver sensitivity can the determined. This is done using the well-known radar range equation below (Balanis, 2015), which defines the relationship between the measured reflection coefficient, the antenna gain, range to the target, etc.

$$|\Gamma|^2 = G_a^2 \left(\frac{\lambda}{4\pi R_w}\right)^2 \frac{\sigma}{4\pi R_w^2} \qquad 25$$

Note that σ is RCS of the target of interest (i.e., the wire). The required transceiver sensitivity (defined earlier as the minimum detectable reflection coefficient magnitude) can be found by solving for Γ in the above equation while setting the other parameters (i.e., $R_w, G_a, \sigma, \lambda$) to match the desired worst-case scenario. Figure 125 shows a plot of the required transceiver sensitivity (Γ_{min}) to detect a power line at a distance of up to 21 meters and at K-band (18-26.5 GHz) vs. the antenna gain.

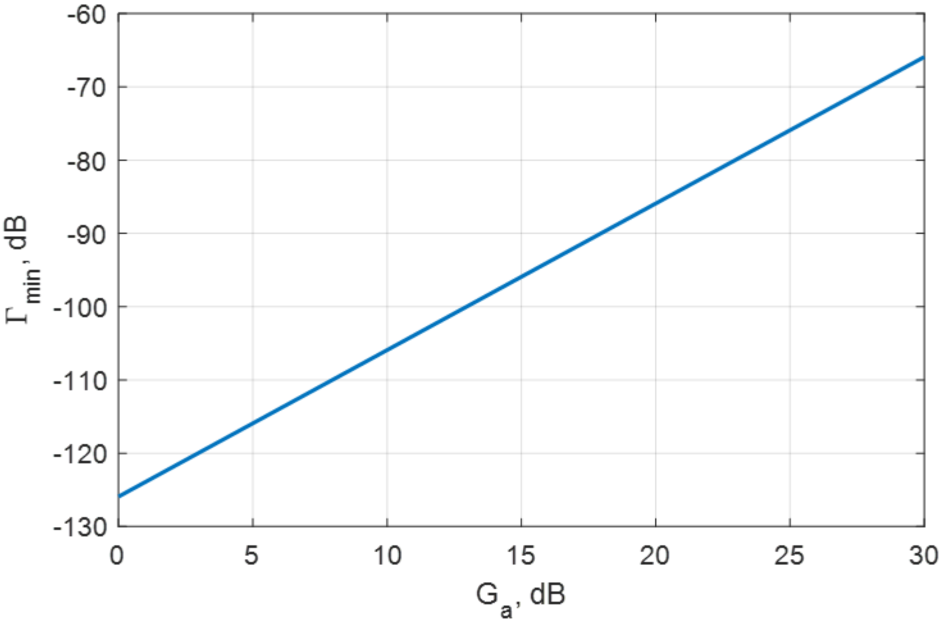


Figure 125. Plot of required transceiver sensitivity (Γ_min) to detect power line at distance of up to 21 meters vs. antenna gain

Note that the data in Figure 125 gives the transceiver requirements for detecting the specific power line used in our specific experiment, when using a K-band (18-26.5 GHz) radar and at a distance of 21 meters. However, the transceiver sensitivity may need to be adjusted slightly depending on the type of power line and would need to be significantly adjusted if the distance to the wire is significantly larger than 21 meters.

Additionally, there are many other specified parameters that can/must be adjusted as needed based on the specific requirements, such as transceiver architecture, the power of the transmitter, power supply requirements, etc. For example, a bistatic radar setup could be used (i.e., separate transmit and receive antennas), which would allow for high-gain and low-noise amplifiers to be added into the signal path. All of these parameters can be adjusted based on additional requirements to be determined in the next phase of this project, as long as the minimum detectable reflection coefficient remains below the specified threshold based on Figure 125.

Additionally, there is a large degree of flexibility in the type of antenna used, as long as it has sufficiently high gain. For example, for a transceiver sensitivity of -80 dB, the antenna gain should be close to 23 dB. Typically, the higher the gain of an antenna, the larger it is and the narrower the beamwidth. For example, the K-band (18-26.5 GHz) horn antenna shown in Figure 121 has a gain of 25 dB and a beamwidth of around 10 degrees, and it has dimensions of around 250 mm x 100 mm x 80 mm. However, there are other types of antennas that can provide the same capabilities but with a smaller footprint. For example, a patch antenna 2D array with a similar gain and beamwidth can be created with smaller dimensions (2 mm x 100 mm x 100 mm).

Since there are so many unknown constraints for the radar system that must be determined in future research (e.g., size requirements, power requirements, etc.), the specific components of the system cannot be specified at this time. However, the findings reported in this section provide guidelines as to the absolute minimum system requirements (i.e., antenna gain, transceiver sensitivity) for successful wire detection.

7 Operators' inputs

Building on initial efforts to collect operator input, additional input was solicited from operators across operating environments and types of operations. The survey previously developed and shown in Appendix A was redistributed to gain insight into current wire strike prevention strategies, operator preferences, and prevalence of near misses during operations. This work also continued the examination of current industry practices in wire strike safety technologies from prior phases.

7.1 Wire strike survey methodology

The purpose of this study was to use survey methodology to identify current industry practices and technologies for wire strike prevention, operator input on which prevention options work best, prevalence of wire strike risk, and then use this operator input to inform best practices. The aim was to broadly sample professionals from across operation types and operating environments. We sought professional helicopter operators input on what type of rotor strike prevention devices work best. A brief, approximately 15-minute-long survey (see Appendix A) elicited input from rotorcraft operators on wire strike safety technologies, along with their personal experience and preferences. From this information, a summary was generated of the technologies currently in use and the recommended best practices for wire strike safety based on industry operator input.

A request for revision of the initial research plan was submitted to and approved by the Florida Tech Institutional Review Board to update the researchers, timeframe of the research, and methods of distribution of materials to include email recruitment of participants in addition to distribution of recruitment cards with QR codes at HeliExpo (IRB 20-007 Revision 1). The research was determined exempt under the category of survey procedures that maintain confidentiality and have minimal risk.

The Wire Strike Safety for Lightweight Helicopters was delivered using Qualtrics survey software. Recruitment was in person with business cards containing the QR code, URL, and basic information; by email to the Helicopter Association International (HAI) via the Rotor Daily email listserv; and by using a snowball sampling strategy encouraging participants to also forward the information emailed to them to others to participate. The initial recruitment was at the HeliExpo 2020 conference; however, this was limited due to the COVID-19 pandemic. Recruitment was extended and included HeliExpo 2022 in Dallas, TX, redistribution of the recruitment materials to HAI members in June via the Rotor Daily, face-to-face recruitment while networking with Florida Tech Aeronautics and Flight test engineering researchers, and email recruitment.

7.2 Demographics

A total of 84 participants started the survey; 75 of these respondents consented to participate and completed a majority of the survey questions.

The respondents broadly represented helicopter professionals. All age categories were represented with the majority of respondents (47%) between 25 and 40 years old (Figure 126). Respondents were predominantly pilots (N=65), but participants also included dispatch, flight crew, and other duties. All levels of experience from student pilots to ATP and CFII were represented (Figure 127); however, instrument rated and commercial pilots were most common. Additionally, the participation represented all major rotorcraft operation types (Figure 128).

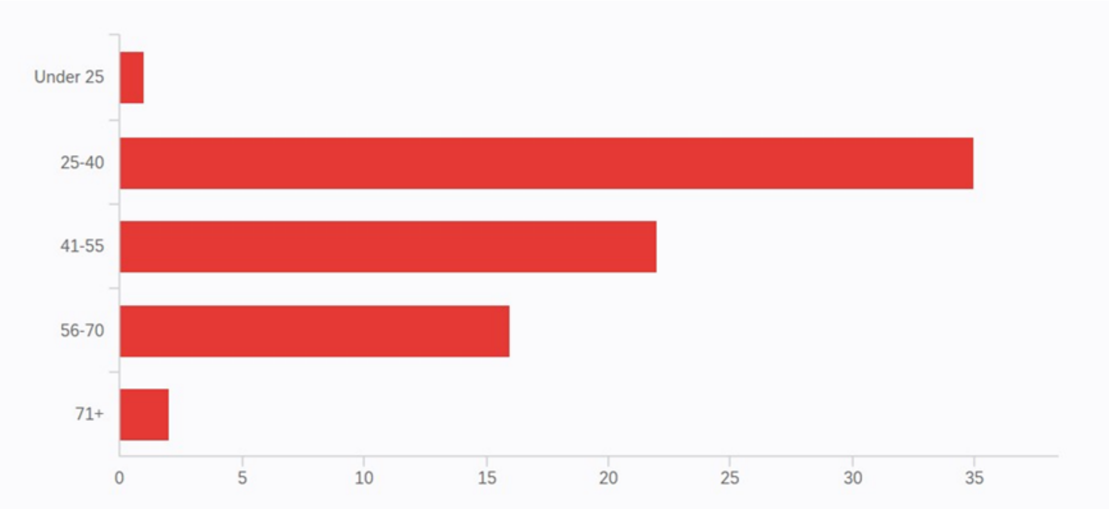


Figure 126. Number of participants by age category

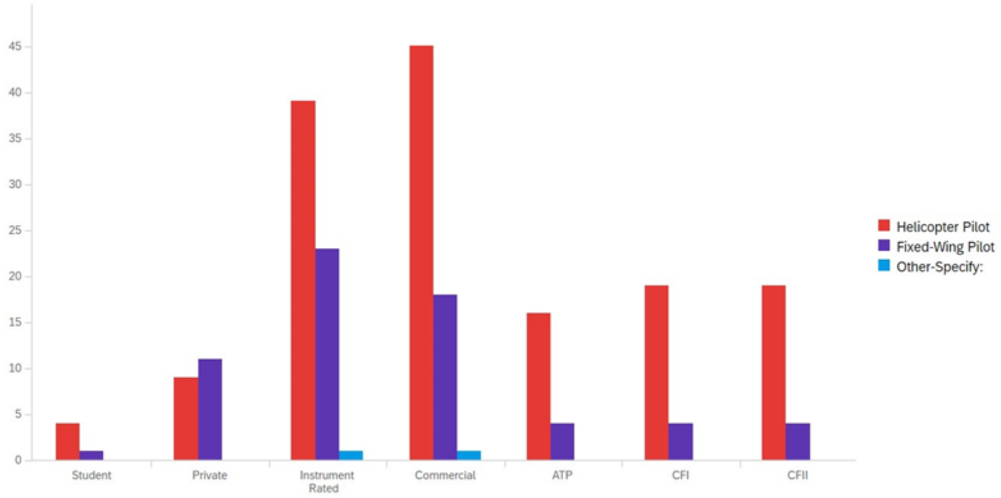


Figure 127. Respondent pilot certificates and ratings

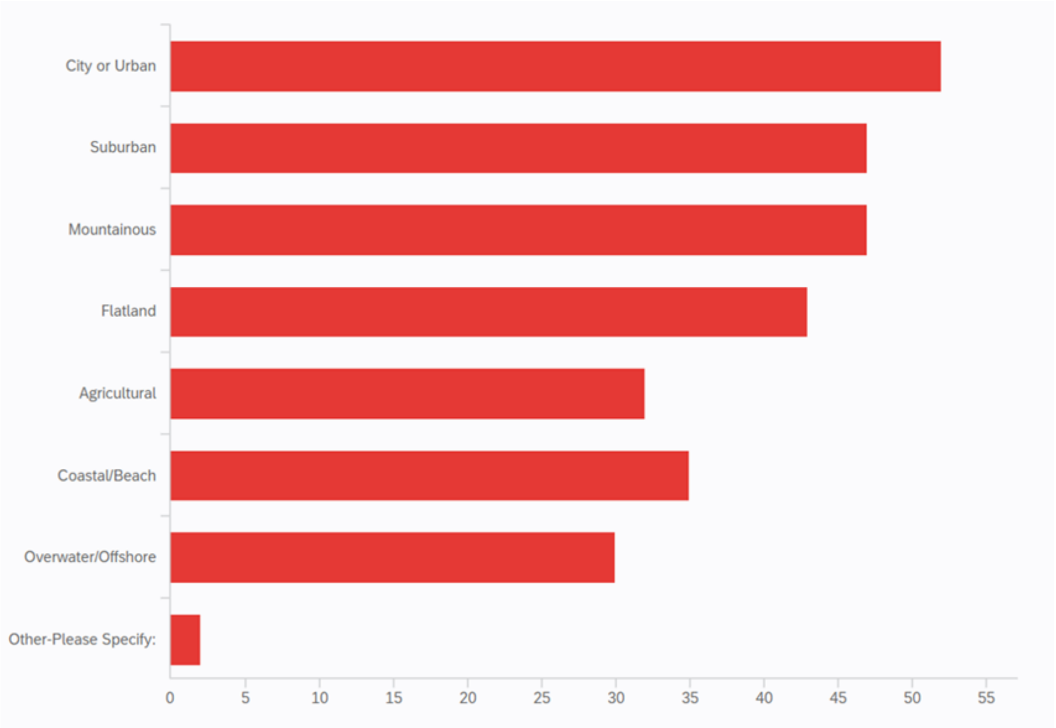


Figure 128. Frequency of experience in operations by domain

Although there were more participants with military experience (35) than any other category, some of these participants also had experience in other types of operations, presumably after their military experience, and there was feedback from all operation types. The location of rotorcraft operations was evenly distributed across all domains (Figure 128), from urban and suburban areas to coastal, mountainous, and agricultural areas.

Based on the demographic information shared by participants, various locations of operations, pilot experience, and types of operations were represented in the study sample.

7.3 Results

Participants indicated concern over wire strikes. While no accidents and only two wire strike-related incidents were reported, half of all participants (49.3%) reported having personally had at least one close call with some form of wire during an operation (Figure 129). Wires and towers with guy wires were most frequently involved in close calls. Participants also reported their preferences for various types of wire strike safety measures (Figure 130).

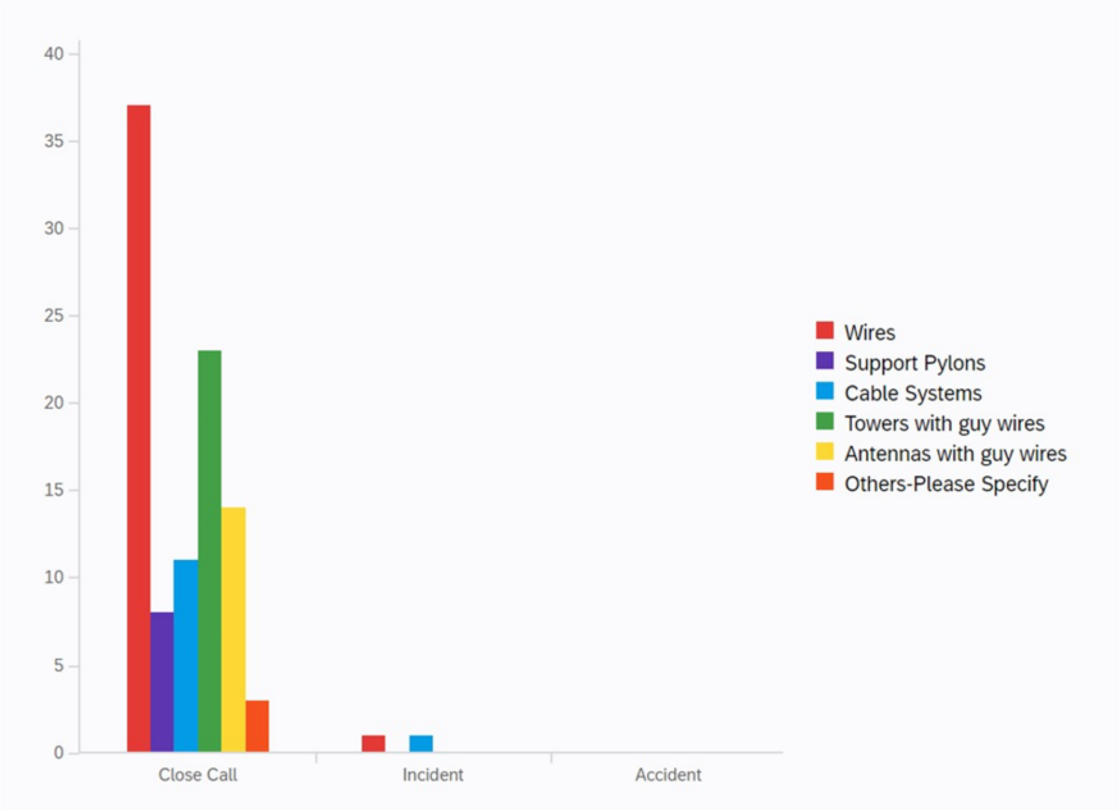


Figure 129. Frequency of respondents who experienced close calls, incidents, and accidents

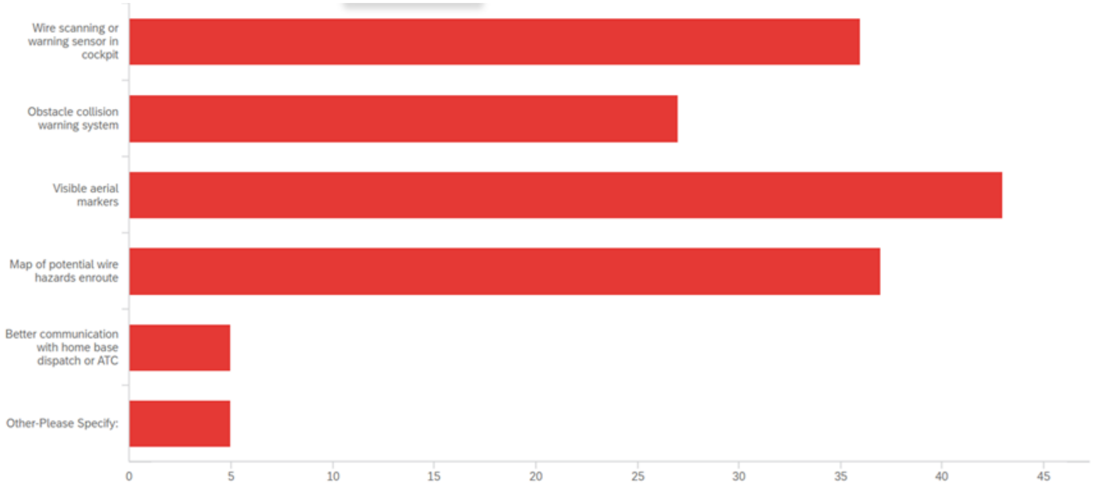


Figure 130. Preference for wire strike prevention methods

Respondents selected visible markers most frequently; however, a map of potential wire hazards and wire scanning/sensing were both selected by almost half of the participants, which supports the need for the work conducted in earlier sections of this project. Obstacle collision warning systems were also a preferred option. Only five participants indicated improved communication with dispatcher or air traffic control as a preferred prevention method.

Open ended responses to questions about what wire strike prevention system(s) or wire cutting devices the participants would be willing to install indicated that participants are very invested in including prevention measures that would improve safety: "Anything that would prevent a strike or catastrophic accident." The most common responses included mechanical wire cutters or standard wire strike prevention system (WSPS) positioned both above and below (N=18), and one response indicated that the wire cutters installed had functioned well. Deflectors were also mentioned by two responses in addition to wire cutters. One response offered the caveat of preferring to install measures that would improve safety but "does not add weight". Several responses mentioned a preference for whichever method is "proven to be safest" (N=3), indicating that operators are interested in methods backed by research. However, five respondents were not certain what method they would prefer. Individual responses of note also included preferences for sensors (2), maps (1), increased situational awareness (1), and that the operator would be "More willing to install systems that identify the wires from either a database or active sensors and give the pilots a chance to avoid them rather than actually trying to cut them" (1).

Specific systems were less frequently named than the general types of systems that operators had experience with. However, the "Dart Aerospace - Blade Cutting Type System" (2), "standard

Bell cutters” (1), “DART MAGELLAN” system (1), and “Garmin GTN 750 w/ Wire Avoidance Installed Cutters” (1) were all named as specific wire strike safety systems that participants were familiar with. Mechanical wire cutters or “no experience” were the most common responses to systems that operators had used.

When asked which wire strike prevention is the best option, participants indicated the systems that they were most familiar with, WSPS, mechanical wire cutter systems being the preference. One response named a specific system: “Magellan systems are the industry standard and should be installed on all helicopters”. However, Figure 130 shows that pilots would prefer maps of wire hazards, sensors, and obstacle avoidance systems as well.

7.4 Recommendations based on operator input survey

The operator input spanned a representative sample of pilot experience and age, as well as most common rotorcraft operation types and domains. Respondents uniformly expressed a preference for including wire strike prevention technologies. They most frequently preferred the technology that they were familiar with, especially wire cutters. However, they rated visual markers, mapping of wire hazards, sensors, and collision avoidance systems as the wire strike safety measures that they would prefer. The pilot preference for using the safest technologies and adding sensors and wire mapping supports the on-going research directions in the earlier sections of this report.

Several respondents were either not at all familiar with the safety options or were uncertain of what they would select. This indicates that even among current helicopter professionals, there is a need to educate pilots about the options for wire strike prevention. This is particularly important given the safety concerns posed by wire strikes, and the high prevalence of close calls, 50% of operators in this study.

8 References

Abadi, M. e. (2016). Tensorflow: A system for large-scale machine learning.

Aerossurance. (2018). *Firefighting Helicopter Wire Strike*. Retrieved 2020, from Aerossurance: http://aerossurance.com/helicopters/firefighting-wire-strike/

Alemi, A. (2016). Improving inception and image classification in tensorflow. *Google Research Blog*.

Balanis, C. A. (2015). *Antenna theory: analysis and design.* John Wiley & Sons.

Berthold, P. a.-J. (2019). A Radar Measurement Model for Extended Object Tracking in Dynamic Scenarios. Paris, France: 0th IEEE Intelligent Vehicles Symposium.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics.* Berlin, Heidelberg: Springer-Verlag.

Brand, S. a. (2012). Multi-core scalable and efficient pathfinding with Parallel Ripple Search.

Burgard, Wolfram, Stachniss, Cyrill, Bennewitz, Maren, & Arras, K. (n.d.). *Introduction to Mobile Robots, Mapping with Known Poses*. (U. o. Freiburg, Editor) Retrieved from https://cs.gmu.edu/~kosecka/cs485/lect17-occupancy-grids.pdf

Burrows, L. T. (1980, 6). Investigation of Helicopter Wire Strike Protection Concepts. *USAAVRADCOM TM-80-D-7.*

Burrows, L. T. (1982). *Verification Testing of a UH-1 Wire Strike Protection System (WSPS).*

Burrows, L. T. (1982). *Verification Testing of an AH-1S Wire Strike Protection System (WSPS).* ARMY AVIATION RESEARCH AND DEVELOPMENT COMMAND ST LOUIS MO.

Case, J. T., Ghasr, M. T., & Zoughi, R. (2011). Optimum two-dimensional uniform spatial sampling for microwave SAR-based NDE imaging systems. *IEEE Transactions on Instrumentation and Measurement, 60*(12), pp. 3806-3815.

Cetin, B. a. (2009). Automated electric utility pole detection from aerial images.

Chandrasekaran, R., Payan, A. P., Collins, K. B., & Mavris, D. N. (2020). Helicopter wire strike protection and prevention devices: Review, challenges, and recommendations. *98.*

Chandrasekaran, R., Payan, A., Collins, K., & Mavris, D. (2019). *A Survey of Wire Strike Prevention and Protection Technologies for Helicopters.* Technical Report, U.S.

Department of Transportation, Federal Aviation Administration. Retrieved from http://actlibrary.tc.faa.gov

Chen, R. T. (1979). *A simplified rotor system mathematical model for piloted flight dynamics simulation.* NASA Ames.

Chollet, F. (2016). Xception: Deep Learning with Depthwise Separable Convolutions.

Cira, C.-I., Alcarria, R., Manso-Callejo, M.-Á., & Serradilla, F. (2020). Framework Based on Nesting of Convolutional Neural Networks to Classify Secondary Roads in High Resolution Aerial Orthoimages. *12*(5).

CustiomPartNet. (n.d.). *Sheet Metal Cutting (Shearing)*. Retrieved 2020, from Custom Part: https://www.custompartnet.com/wu/sheet-metal-shearing

Davis, J. M. (1974). *Rotorcraft Flight Simulation with Aeroelastic Rotor and Improved Aerodynamic Representation.* Army Air Mobility Research and Development Laboratory.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. Miami, FL: IEEE Conference on Computer Vision and Pattern Recognition.

Du, J. (2018). Understanding of object detection based on CNN family and YOLO.

Eisner, J., Funke, S., Herbst, A., Spillner, A., & Storandt, S. (2011, October). Algorithms for Matching and Predicting Trajectories., (pp. 84-95). doi:10.1137/1.9781611972917.9

Everingham, M., Van~Gool, L., Williams, C., Winn, J., & Zisserman, A. (2012). *The PASCAL Visual Object Classes Challenge 2012 VOC2012 Results*. Retrieved from http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html

Fryskowska, A. (2019). Improvement of 3D Power Line Extraction from Multiple Low-Cost UAV Imagery Using Wavelet Analysis. *19*(3).

Garg, L. a. (2019). Land Use Land Cover Classification from Satellite Imagery using mUnet: A Modified Unet Architecture.

Gershenson, D., Rohrer, B., & Lerner, A. (2019). *A New Predictive Model for More Accurate Electrical Grid Mapping.* Tech. rep., Facebook Engineering. Retrieved from https://engineering.fb.com/connectivity/electrical-grid-mapping/

Goshi, Darren and Case, Timothy and McKitterick, John and Bui, Long. (2012). *8361.*

Gunning, M., Gershenson, D., Rohrer, B., & Lerner, A. (2019). *A New Predictive Model for More Accurate Electrical Grid Mapping.* Facebook Engineering.

Hoffman, P., Streit, D., Gilstrap, M., Amin, S., & DeCorla-Souza, K. (2015). *United States Electricity Industry Primer.* Tech. rep., U.S. Department of Energy, Office of Electricity Delivery and Energy Reliability.

Irvine, M. (1981). *Cable Structures.* The MIT Press.

Kim, M. a. (2011). The Network Completion Problem: Inferring Missing Nodes and Edges in Networks.

Kleinberg, J., & Tardos, E. (2005). Algorithm Design. Boston, MA, United States: Addison-Wesley Longman Publishing Co., Inc.

Krizhevsky, A. a. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems.* 1097--1105.

Krylov, V. A., Kenny, E., & Dahyot, R. (2018). Automatic discovery and geotagging of objects from street view imagery. *Remote Sensing, 10*, 661.

Lundquist, C. a. (2010). Road intensity based mapping using radar measurements with a probability hypothesis density filter. *59*(4).

Ma, Q., Goshi, D., Bui, L., & Sun, M.-T. (2015). Robust power line detection with particle-filter-based tracking in radar video. (4).

Ma, Q., Goshi, D., Shih, Y., & Sun, M.-T. (2011). An Algorithm for Power Line Detection and Warning Based on a Millimeter-Wave Radar Video. *20.*

Magellan Aerospace. (n.d.). *Magellan Aerospace WSPS*. Retrieved 2020, from Magellan Aerospace: https://magellan.aero/product/wsps/

Mahvash, M., Voo, L., Kim, D., Jeung, K., Wainer, J., & Okamura, A. (2008). Modeling the Forces of Cutting With Scissors. *55.*

Matikainen, L., Lehtomäki, M., Ahokas, E., Hyyppä, J., Karjalainen, M., Jaakkola, A., . . . Heinonen, T. (2016). Remote sensing methods for power line corridor surveys. *119.*

McGlinchy, J. a. (2019). Application of UNet Fully Convolutional Neural Network to Impervious Surface Segmentation in Urban Environment from High Resolution Satellite Imagery.

Meissner, S. (1997). Mechanics of a shear cutting process.

New York State GIS. (2022). *NYS Interactive Mapping Gateway*. Retrieved from NYS Interactive Mapping Gateway: http://gis.ny.gov/gateway/mg/

QGIS Development Team. (2022). *QGIS Geographic Information System.* Retrieved from https://www.qgis.org

Ren, S. e. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks.

Richards, M. A., Scheer, J. A., & Holm, W. A. (2010). *Principles of Modern Radar.* SciTech Publishing, Inc.

Ronneberger, O. a. (2015). U-net: Convolutional networks for biomedical image segmentation.

Saha, M. a. (2018). Her2net: A deep framework for semantic segmentation and classification of cell membranes and nuclei in breast cancer evaluation.

Sahin, I., & Kemal, O. (2014). A comparative evaluation of well-known feature detectors and descriptors. *3*(1).

Sarabandi, K., & Monsoo, P. (1994). Power lines: Radar measurements and detection algorithm for polarimetric SAR images. *IEEE Transactions on Aerospace and Electronic Systems, 30*(2), pp. 632-643.

Sarabandi, K., & Monsoo, P. (1999). Millimeter-wave radar phenomenology of power lines and a polarimetric detection algorithm. *IEEE Transactions on Antennas and Propagation, 47*(12), pp. 1807-1813.

Sarabandi, K., & Monsoo, P. (2000). Extraction of power line maps from millimeter-wave polarimetric SAR images. *IEEE Transactions on Antennas and Propagation, 48*(12), pp. 1802-1809.

Stadtmueller, T. (2018). *Rotorcraft Wire Strike Data.* Tech. rep., Federal Aviation Administration.

Storandt, J. E. (2019). Algorithms for Matching and Predicting Trajectories. In *2011 Proceedings of the Thirteenth Workshop on Algorithm Engineering and Experiments ({ALENEX}).* 84--95.

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. *Thirty-first AAAI conference on artificial intelligence.*

Tang, Q., Wang, Z., Majumdar, A., & Rajagopal, R. (2019). Fine-Grained Distribution Grid Mapping Using Street View Imagery. *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, BC, Canada*, (pp. 8–14).

The World Bank. (2018). *Machine Learning for High Resolution High Voltage Grid Mapping.* Development Seed. ESMAP. Retrieved from http://devseed.com/ml-grid-docs/

Thrun, S. (n.d.). *Learning Occupancy Grids with Forward Sensor Models*. (C. M. University, Editor) Retrieved from https://www.cs.cmu.edu/~thrun/papers/thrun.occ-journal.pdf

Treml, M. a.-M. (2016). Speeding up semantic segmentation for autonomous driving.

Tsagkatakis, G. a. (2019). Survey of Deep-Learning Approaches for Remote Sensing Observation Enhancement. *19*(10.3390/s19183929).

U.S. Department of Energy, Office of Electricity Delivery and Energy Reliability. (2015). *United States Electricity Industry Primer.*

U.S. Department of Homeland Security. (2020). Homeland Infrastructure Foundation-Level Data (HIFLD). Retrieved from https://hifld-geoplatform.opendata.arcgis.com/

Wagner, S. K. (2007). Computing Many-to-Many Shortest Paths Using Highway Hierarchies. In *2007 Proceedings of the Ninth Workshop on Algorithm Engineering and Experiments ({ALENEX}).* Society for Industrial and Applied Mathematics.

Warmick, M., & Hoffman, M. (2016). *Electricity Distribution System Baseline Report for DOE Quadrennial Energy Review.* Pacific Northwest National Laboratory.

Williams, J. P. (2016). Fundamentals of cutting. Retrieved from https://www.semanticscholar.org/paper/Fundamentals-of-cutting-Williams-Patel/fb5ec2b9a03a2f2e670aad83678536073e3d876b

Wu, W., & Cao, X. (2016). Mechanics model and its equation of wire rope based on elastic thin rod theory. *102-103*.

Xu, B., Dong, P., Zhang, J., & Zhang, S. (2014). A three dimensional approach to model steel wire ropes used in high energy absorber apparatus. *51*(25).

Yan, G., Li, C., Zhou, G., Zhang, W., & Li, X. (2007). Automatic Extraction of Power Lines From Aerial Images. *4*(3).

Zain, A., Baseer, S., & Munir, M. (2018). Transmission Lines Monitoring from Satellite Images.

Zhang, W., Witharana, C., Li, W., Zhang, C., Li, X., & Parent, J. (2018). Using deep learning to identify utility poles with crossarms and estimate their locations from google street view images. *Sensors, 18*, 2484.

Zhang, Y., Yuan, X., Li, W., & Chen, S. (2017). Automatic Power Line Inspection Using UAV Images. *9*(8).

Zhou, L. a. (2018). D-LinkNet: LinkNet With Pretrained Encoder and Dilated Convolution for High Resolution Satellite Imagery Road Extraction.

Zhu, X., Tuia, D., Mou, L., Zia, G., Zhang, L., Xu, F., & Fraunkdorfer, F. (2017). Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources. *5*(4).

A Operator input survey

Wire Strike Safety for Lightweight Helicopters

Start of Block: Wire Strike Safety

CONSENT DOCUMENT FOR ENROLLING ADULT PARTICIPANTS IN A RESEARCH STUDY

Florida Institute of Technology
College of Aeronautics
Informed Consent

Please read this consent document carefully before you decide to participate in this study.

Study Title: Wire Strike Safety for Lightweight helicopters survey for the FAA Center of Excellence for GA

Purpose of the Study: The research seeks to reduce the number of helicopter wire strike fatalities. Participants chosen are rotorcraft professionals active in the helicopter industry.

Procedures: Participation will be by completing a brief survey about your own experience lasting about 15 minutes.

Potential Risks of Participating: No risks are associated with this study beyond that normally experienced when describing past experiences with a colleague.

Potential Benefits of Participating: You are not likely to benefit in any immediate way from joining this study. However, your participation in this study may assist researchers in understanding how to improve helicopter safety.

Confidentiality: Your identity will be kept confidential to the extent provided by law. Your information will be assigned a code number, instead of any personally identifying information. The list connecting your name to this number will be kept in a locked file in the FIT College of Aeronautics. When the study is completed and the data has been analyzed, the list will be destroyed. Your name will not be used in any report.

Voluntary participation: Your participation in this study is completely voluntary. There is no penalty for not participating. No compensation for your time is available. You may also refuse to answer any of the questions we ask you.

Right to withdraw from the study: You have the right to withdraw from the study at any time without consequence.
Whom to contact if you have questions about the study: Dr. Brooke Wheeler, FIT College of Aeronautics,
Email: bwheeler@fit.edu

Whom to contact about your rights as a research participant in the study:
Dr. Jignya Patel, IRB Chair
Florida Institute of Technology
150 West University Blvd.
Melbourne, FL 32901
Email: jpatel@fit.edu Phone: 321.674.7391

Agreement:
I have read the procedure described above. I voluntarily agree to participate in the procedure and have received a copy of this description.

○ I Agree to Participate in the Study (1)

○ I Do Not Agree to Participate in this Survey (2)

Skip To: 1. If CONSENT DOCUMENT FOR ENROLLING ADULT PARTICIPANTS IN A RESEARCH STUDY Florida Institute of Techno... = I Agree to Participate in the Study

Skip To: End of Survey If CONSENT DOCUMENT FOR ENROLLING ADULT PARTICIPANTS IN A RESEARCH STUDY Florida Institute of Techno... = I Do Not Agree to Participate in this Survey

1. 1 What is your age range?

☐ Under 25 ☐ 25-40 ☐ 41-55 ☐ 56-70 ☐ 71+

2. What is your duty and how long have you been in that position?

☐ Pilot ____ Years
☐ Crew ____ Years
☐ Dispatcher ____ Years
☐ Other – Specify: __________________ ____ Years

3. If you are a pilot, what is the extent of your piloting experience?

Please select all that apply and write the approximate corresponding flight time

☐ **Helicopter Pilo**t: ____ Hours ____ Years
☐ Student ☐ Private ☐ Commercial ☐ ATP ☐ CFI ☐ CFII

☐ **Fixed-Wing Pilot**: ____ Hours ____ Years
☐ Student ☐ Private ☐ Commercial ☐ ATP ☐ CFI ☐ CFII

☐ **Instrument rated?** ☐ Yes ☐ No
☐ Instrument time (actual) ____ Hours
☐ Instrument time (simulated) ____ Hours
☐ **Other – Specify: ______________________** ____ Hours ____ Years

4. If you are a helicopter pilot, what types of rotorcraft do you have experience with?
Please select all that apply and write the approximate corresponding flight time

Engine Type

☐ Single Engine ____ Hours
☐ Twin Engine ____ Hours
☐ Turbine Powered ____ Hours
☐ Piston Powered ____ Hours

☐ Other – Specify: ______________________ ____Hours

Rotor Type

☐ Single Main Rotor ____ Hours
☐ Tandem rotors ____ Hours
☐ Coaxial rotors ____ Hours
☐ Intermeshing rotors ____ Hours
☐ Other – Specify: ______________________ ____ Hours

5. In what operational domain do you have helicopter aviation experience?

Please select all that apply and write the approximate corresponding flight time or years.

☐ Construction or Gas & Electric utilities ____ Hours ____ Years
☐ Offshore Oil Exploration ____ Hours ____ Years
☐ Emergency Medical Services or Air Ambulance ____ Hours ____ Years
☐ Search and Rescue ____ Hours ____ Years
☐ Law enforcement or Aerial Observation/Surveillance____ Hours ____ Years
☐ Commercial Air Tour or Air Taxi ____ Hours ____ Years
☐ Military or Governmental ____ Hours ____ Years
☐ Private/Recreational ____ Hours ____ Years
☐ Other – Please Specify: ________________ ____ Hours ____ Years

6. What type(s) of terrain do you normally fly in or help helicopter pilots and crew fly in?

Please check all that apply

☐ City or Urban
☐ Suburban
☐ Mountainous
☐ Flatland
☐ Agricultural
☐ Coastal or Beach
☐ Over Water
☐ Other – Please Specify: _____________________

7. What type(s) of conditions do you normally fly in or help pilots and crew fly in?

Please check all that apply

☐ Daytime VFR
☐ Nighttime VFR
☐ Instrument Flight Rules (IFR)
☐ Flight with Night Vision Goggles/Devices (NVG/NVD)
☐ Other – Please Specify: _____________________

8. What wire strike protection systems or wire cutting devices are you familiar with? Please specify helicopter make and model, type of device and other comments.

☐ Helicopter Make Specify: ______________________
☐ Helicopter Model Specify: ______________________
☐ Type of device Specify: ______________________
☐ Other comments (please add below)

9. What type of wire strike prevention system(s) or wire cutting devices would you be willing to install on your helicopter or which do you prefer? Please comment below.

____

____

____

____

____

10. Do you have any recommendations or best practices on which wire strike devices to use or which device might work best? Please comment below

____

____

____

____

____

11. **The following questions may contain sensitive information. You may choose to answer them or not.**

Have you ever encountered, or come close to encountering, a dangerous wire strike safety condition or event involving one or more of the hazards listed below? Did any of them result in an aviation incident or accident? ***Please select all that apply***

	Close Call (1)	Incident (2)	Accident (3)
Wires (1)	☐	☐	☐
Support Pylons (2)	☐	☐	☐
Cable Systems (3)	☐	☐	☐
Towers with guy wires (4)	☐	☐	☐
Antennas with guy wires (5)	☐	☐	☐
Others-Please Specify (6)	☐	☐	☐

12. If you selected any of the previous question's entries, please briefly explain the circumstances including the environment in which the wire(s) or supporting structures were encountered

____

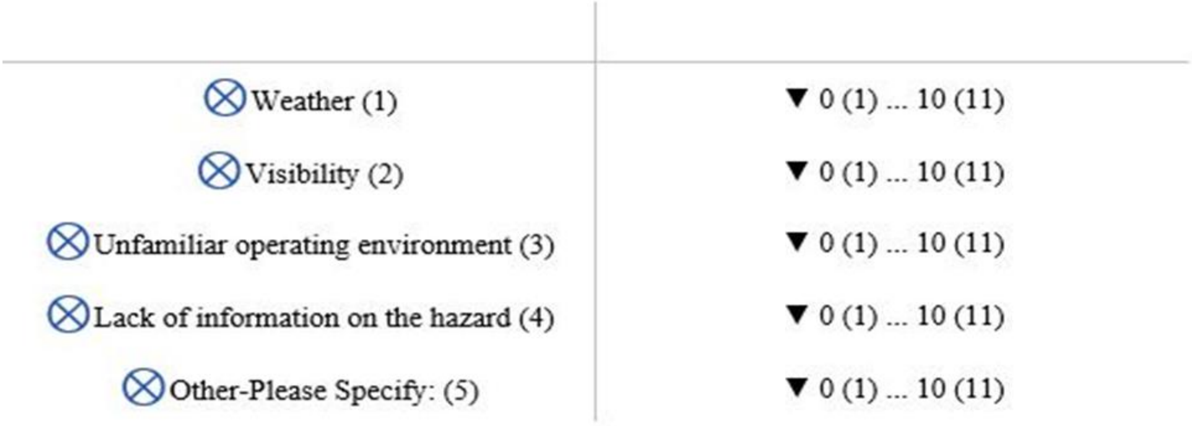
____

____

____

____

13. For the helicopter operations you deal with, on a scale from 0 (never) to 10 (always), how often would you say the following factors had an impact on the wire safety event?



⊗ Weather (1)	▼ 0 (1) ... 10 (11)
⊗ Visibility (2)	▼ 0 (1) ... 10 (11)
⊗ Unfamiliar operating environment (3)	▼ 0 (1) ... 10 (11)
⊗ Lack of information on the hazard (4)	▼ 0 (1) ... 10 (11)
⊗ Other-Please Specify: (5)	▼ 0 (1) ... 10 (11)

14. What type of intervention(s) do you feel could have prevented you from getting into a dangerous wire strike situation? A short list is provided for your review. ***Please select all that apply/add comments***

☐ Wire scanning or warning sensor in cockpit (1) ____

☐ Obstacle collision warning system (2) ____

☐ Visible aerial markers (3) ____

☐ Map of potential wire hazards enroute (4) ____

☐ Better communication with home base dispatch or ATC (5) ____

☐ Other-Please Specify: (6) ____

15. Would you be willing to participate in an interview with a researcher to gain a better understanding of the wire strike related event you were involved in (related to questions 11-14)?

○ Yes (1)

○ No (2)

---

16. Would you be interested in continuing your participation in this study?

○ Yes (1)

○ No (2)

---

17. If you answered "Yes" to either of the previous two questions, please leave your contact information below (either one or both is sufficient) so that we may reach out to you in the near future!

○ Email (1) ____

○ Phone (2) ____

End of Block: Wire Strike Safety