Vision-Based Multi-Wire Detection and Tracking for UAV Wire Approach

Tyler Harp

CMU-RI-TR-25-100

December 15th, 2025

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Professor Sebastian Scherer, *Chair*
Professor Wennie Tabib
Mohammadreza Mousaei

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Robotics.

To my family, whose unwavering support and love have been my foundation and strength, and who gave me the belief to embark on this journey.

Abstract

Reliable detection and tracking of power lines is critical for enabling under-wire UAV approach and inductive power-line charging to extend UAV range. However, wires are thin, featureless, and visually ambiguous structures that challenge traditional computer vision methods and degrade depth estimation accuracy. To address these challenges, this thesis presents a fully passive, camera-only multi-wire detection and tracking algorithm that operates in real time on lightweight onboard compute using only stereo RGB imagery.

The proposed framework integrates a lightweight classical vision pipeline with three-dimensional geometric reasoning and a multi-stage filtering process to produce robust wire instance detections. A complementary oriented object detection model is trained using labels generated by the classical pipeline, leveraging its fine-tuned geometric outputs to improve resilience in challenging visual conditions. To track individual wire instances across frames, we introduce a Kalman-filter-based tracking architecture that estimates both wire orientation and per-wire positional state while remaining robust to wire detection outliers and vehicle pose drift. The system is further expanded by testing a wire positional servoing approach using the tracked wire instances in simulation and is validated across a diverse range of data sources, including simulation, indoor testing, and outdoor flight evaluations.

Acknowledgments

I want to give a huge thank you to Basti and Steve, whose support and guidance throughout my masters has been vital to this work. Thank you for allowing me to tinker with drones around the clock, it truly has been a dream come true.

I would also like to give a special thank you to my other committee members, Wennie and Mohammad, who have shared career-long lessons with me in such a short period and have been incredibly generous with their time.

Thank you to my friends and family. Without their support, which came in so many different forms, I would not have been able to complete the work presented here.

And to the systems engineers, scientists, and fellow researchers in the AirLab: it has been amazing to learn from you and I have grown immensely through your knowledge.

Funding

This work was sponsored by the ICWERX group.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem Space

Unmanned Aerial Vehicles (UAVs) have become indispensable tools across a wide range of applications, including infrastructure inspection, defense, precision agriculture, and disaster response. As these systems assume increasingly complex and autonomous roles, a fundamental limitation persists: the restricted flight time imposed by current battery technologies. This constraint is especially problematic in remote, large-scale, or continuous-operation scenarios, where returning to a base station for manual recharging is impractical or impossible. Extending the effective operational range and endurance of UAVs is therefore one of the central technological challenges in modern aerial robotics. Overcoming this limitation would dramatically broaden the range of feasible UAV missions, enabling operations that are currently constrained by the need for frequent recharging at fixed geographic locations.

A promising approach to addressing this challenge is powerline inductive charging, which allows UAVs to recharge by leveraging the electrical infrastructure already present throughout urban, suburban, and rural environments. By exploiting the alternating magnetic fields produced by high-voltage transmission lines, a current can be induced in a specially designed coil mounted on the UAV and used to recharge its onboard battery. This method requires the UAV to securely clamp around the wire and approach it with high precision to achieve efficient energy transfer. However, enabling inductive charging in flight demands exceptionally accurate wire localization, tracking, and approach, all under diverse environmental and sensing conditions. As such, the ability to reliably detect and track power lines becomes a critical enabling capability and the key focus of this thesis.

1.2 Challenges

Detecting wires in visual data presents unique challenges due to their thinness, lack of distinctive features, and tendency to appear in clusters or bundles. Unlike conventional objects with texture, color variation, or well-defined shapes, wires are slender structures with minimal visual cues, making them difficult for standard computer vision algorithms to detect and track.

Their thinness makes geometry and depth estimation challenging. Wires occupy only a few pixels in typical camera resolutions, causing them to be easily lost during downsampling or overlooked by convolutional neural networks that favor wider patterns. Their small image footprint also increases sensitivity to noise and motion blur.

Wires are also largely featureless, offering few salient keypoints. This limits the effectiveness of traditional feature-based methods, which rely on textured or visually rich regions to establish correspondences or maintain stable visual anchors across frames.

Finally, wires also frequently appear in clusters or bundles, where multiple thin, visually similar strands overlap or run parallel. Segmentation methods struggle to distinguish individual wires, and intersections, occlusions, or perspective merges further increase ambiguity, complicating instance separation and geometry inference.

Together, these factors make wire detection and tracking fundamentally challenging. Effective methods must reason about the continuity and geometry of fine linear structures rather than relying solely on conventional appearance-based cues.

1.3 Contributions

In this thesis, we present a robust, multi-wire, vision-based system capable of both detecting and tracking power lines using only onboard RGB cameras. Our method addresses the inherent challenges of relying exclusively on passive visual sensing, which demands advanced algorithms for precise wire localization and consistent temporal tracking across diverse environmental conditions. The framework is designed to generalize across varying wire configurations, backgrounds, and real world scenarios. Through this work, we evaluate the algorithm in a wide range of environments, including indoor, outdoor, and simulated settings.

In addition to the classical vision approach, we introduce a learning based wire detection module trained on wire annotations generated automatically by our system. This modern deep learning detector complements the traditional perception pipeline by improving robustness under challenging visual conditions and enhancing overall tracking stability. Together, these components form a modular perception framework that enables reliable multiwire detection and tracking in both simulation and real world environments.

Lastly, we demonstrate that this algorithm operates accurately and reliably when compared with existing approaches, many of which struggle to generalize due to their reliance on active sensing or restrictive environmental assumptions. We specifically validate the depth accuracy and real time performance of our method, verifying that it is suitable for deployment in real world scenarios and adaptable to a wide range of conditions.

Chapter 2

Background

When considering the use of existing powerline infrastructure as a charging solution for UAV systems, several interconnected challenges arise. Figure 2.1 provides a visual overview of these obstacles in the autonomous wire charging pipeline. The first challenge is reliably localizing wire infrastructure from long, predominantly horizontal viewpoints. The second involves planning around this infrastructure and positioning the UAV in safe, close proximity to the wires. The final challenge concerns the approach itself, specifically how to physically engage with the wire to establish a charging connection.

This thesis focuses primarily on the last approach stage: developing a robust method for tracking individual wires and approaching to a selected target wire to begin charging. The guiding question, then, is how we can design a system capable of achieving this reliably under real world constraints.

2.1 Existing Wire Tracking Methods

A wide range of sensing and perception strategies have been explored for detecting and tracking overhead wires and other cylindrical structures, driven by the need for UAVs to interact safely and accurately with linear features. Numerous sensor choices and tracking techniques have been proposed, creating a broad landscape of possible approaches. These sensors generally fall into two main categories: depth based sensors and vision based sensors. Depth based sensors commonly include solid state or rotary

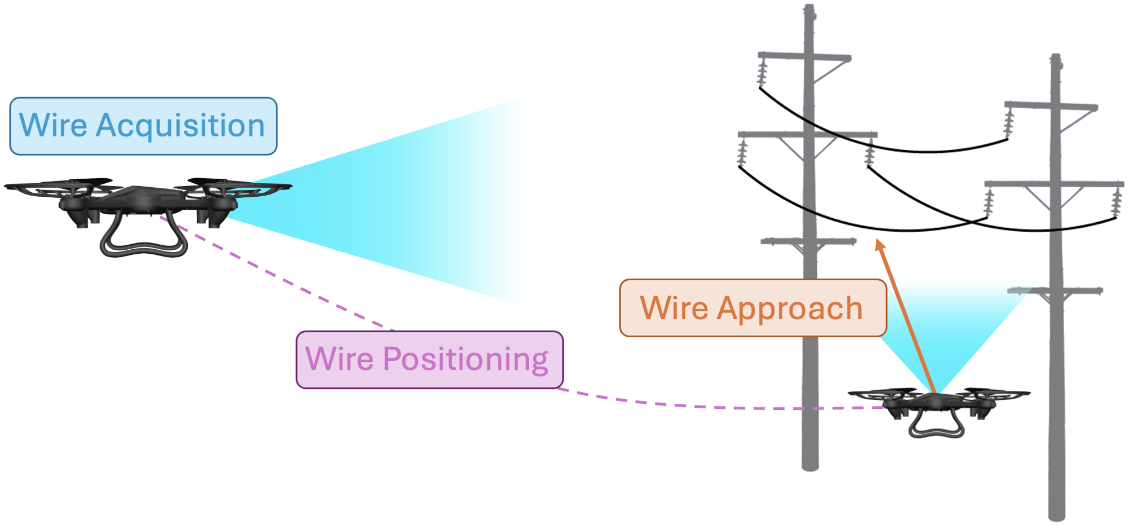


Figure 2.1: Breakdown of the challenges involved in autonomous UAV wire charging

LiDAR and radar systems, while vision based sensors typically include RGB, thermal, and stereo cameras capable of extracting depth. The following sections outline how these two sensor categories have been used in existing wire tracking methods and describe the limitations associated with each.

2.1.1 Depth Sensor Based Tracking

Several wire tracking systems rely primarily on active range sensing to deduce wire depth and orientation. For example, LOCATOR [5] employs two multi-segment LiDAR units to estimate the cable location and angle above. While effective for single-wire scenarios, their usefulness is constrained by the sparsity of LiDAR returns on thin wires. In practice, perception range is limited to only a few meters, and the density of measurements is insufficient for distinguishing multiple wires or maintaining fine grained estimation of a wire's pose. Because these approaches also rely on active sensors, they introduce additional weight and cost. Overall, depth-based methods offer useful geometric cues but do not apply well to lightweight platforms, long detection distances, and cluttered multi-wire environments.

2.1.2 Vision Based Tracking

A substantial portion of prior work on tracking linear structures relies exclusively on camera based perception, using either RGB imagery or RGB-D sensors. These methods span several related domains, including visual servoing of cylindrical objects, pipe tracking for inspection tasks, and wire detection supported by stereo or RGB-D depth estimation.

Some camera only approaches focused primarily on larger cylindrical targets. For example, [12] demonstrated autonomous landing on outdoor pipes using an RGB-D stereo camera, segmenting the pipe and exploiting the strong geometric consistency and reliable depth cues associated with large diameter structures. Similarly, visual servoing methods such as [14] depend on extracting stable edges, textures, or other geometric features to guide UAV interaction with cylindrical objects. However, these approaches typically require a priori knowledge of the cylinder's width or the number of poles in view to recover a consistent pose estimate.

More recent vision based work targeting wire specific perception attempts to localize wires using RGB-D sensors. For instance, [7] combines RGB imagery with stereo depth maps to estimate the 3D position of an individual wire by using depth gradient cues. However, their method, like most pure vision approaches, must assume that only a single wire is present in the scene where violation of this assumption leads to failure.

2.1.3 Sensor Fusion Based Tracking

More recent systems incorporate both depth based and vision based sensors to leverage the strengths of multiple modalities. Joint work from [9] and [4] combines radar measurements with an FPGA accelerated vision front end to detect high voltage transmission lines at moderate distances. LineDrone [11] integrates a monocular camera with a 2D LiDAR to reconstruct the local geometry of powerlines. However, incorporating multiple heterogeneous sensor sources increases the need for specialized hardware and additional computational pipelines, which in turn adds system complexity. This added complexity reduces the lightweight, simplistic nature desirable for small UAS platforms, and the reliance on rigid compute architectures makes such systems less adaptable.

2.2 System Design Criteria

To evaluate existing methods, we first established the core requirements our system must satisfy. It must be ubiquitous, functioning across industries and mission types without specialized infrastructure; multi-wire capable, able to distinguish and select a single wire among many; lightweight, suitable for small UAVs; accurate, ensuring estimation errors do not lead to unsafe approaches; and real time, allowing the UAV to react immediately to changing conditions.

Guided by these requirements, we wanted to design a system centered on passive sensing with a single onboard camera and a modular compute stack to support broad deployment. The perception pipeline performs wire instance detection and tracking to handle multiple wires, while all computation runs on compact edge hardware using only camera data to keep the system lightweight. Accuracy is achieved through explicit error evaluation and temporal smoothing, and real-time operation is maintained by processing every camera frame and continuously predicting the target wire's position. This combination enables safe, precise wire approach under real-world constraints.

2.3 Sensor Selection

When evaluating our system requirements, we sought a sensing approach that satisfied all these stipulations, and a single stereo camera emerged as the strongest fit. It provides multiple sensing capabilities within a lightweight, passive package: its dual modality supplies both depth and visual information from the same sensor, enabling reliable wire localization without additional hardware. Unlike sparse range sensors, stereo offers dense depth across the entire field of view, which is crucial for detecting and tracking thin, low-texture structures such as powerlines. As a single passive sensor, it keeps the system simple, power efficient, and easy to deploy across varied missions and platforms, supporting the ubiquity and multi-mission flexibility needed for real world UAV wire approach operations.

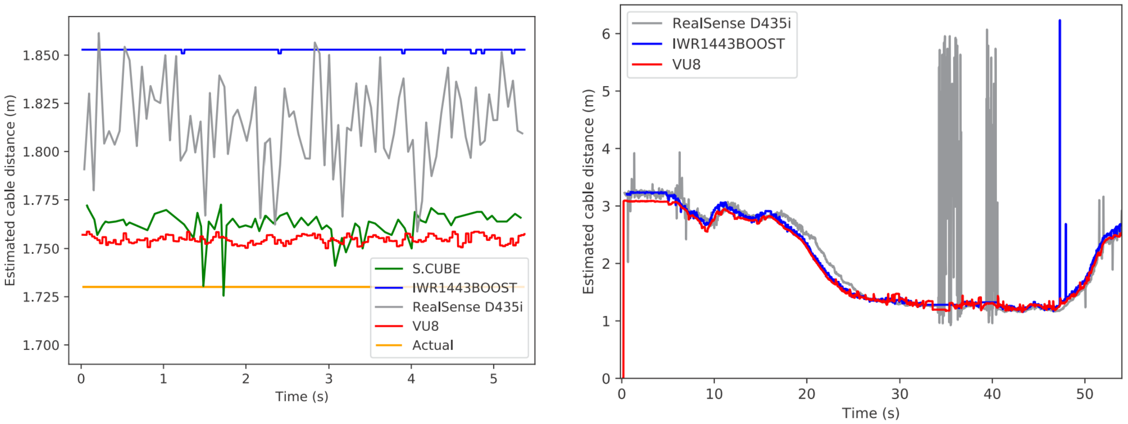


Figure 2.2: Left: Estimated detection distance of a static overhead wire using different sensors [8], Right: Dynamic detection of wire distance over time across different sensors [8]

2.4 Sensing Challenges

However, if a stereo camera were truly the ideal choice for building a general and simplified system, it would be more prevalent in existing approaches. In practice, the advantages of stereo sensing for UAV applications are often offset by its susceptibility to noise and systematic depth biases. As shown in Figure 2.2, adapted from [8], stereo depth estimates exhibit substantially higher variance compared to alternative sensing modalities. The grey trace in particular illustrates the pronounced noise characteristics that can degrade distance estimation accuracy when observing thin structures such as wires.

Additionally, Figure 2.3 illustrates several characteristic failure modes of learning based dense depth estimation. Neural stereo depth methods often introduce biases that manifest as surface waviness and frame to frame inconsistency, complicating reliable wire localization. These models also tend to blur or smear fine edges, particularly in thin structures such as wires, as documented in [15], due to limitations inherent in their loss formulations. Given these challenges, the perception stack must explicitly compensate for the shortcomings of stereo based depth. The following sections outline the design of this perception stack and the strategies employed to address these issues.

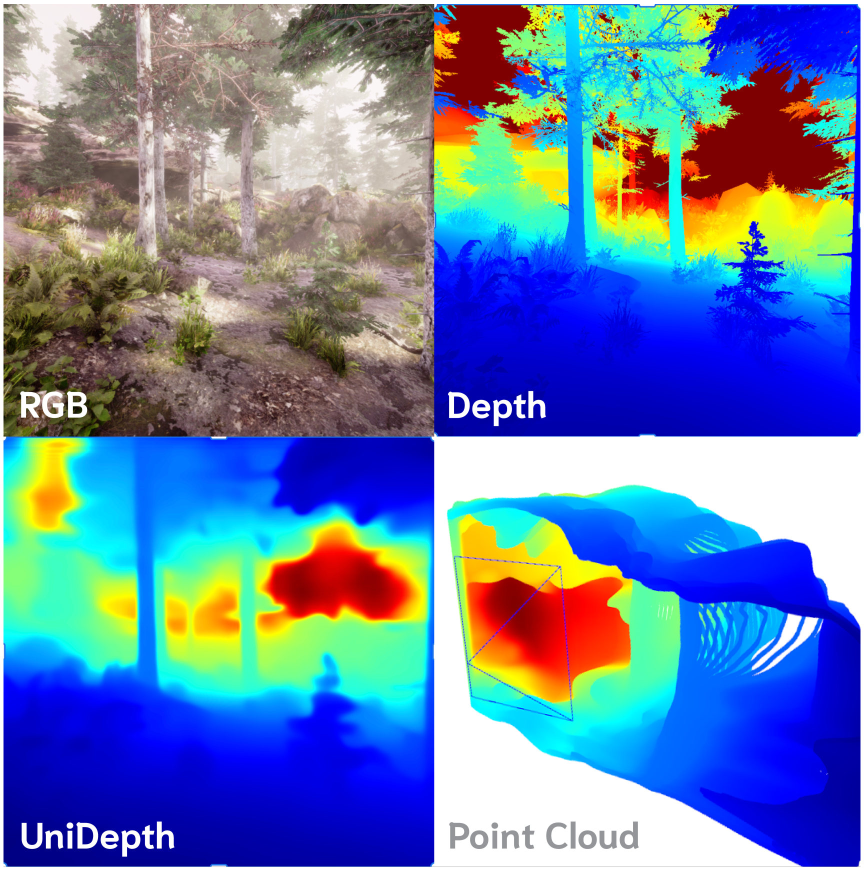


Figure 2.3: Comparison of ground truth depth with the biases and smoothing introduced by neural dense depth estimation

Chapter 3

Wire Approach Methodology

3.1 System Overview

The perception system for wire monitoring consists of three primary components: wire detection, wire tracking, and wire approach. Before entering this pipeline, the stereo image pair is processed by segmentation and depth estimation algorithms to produce the multimodal data used throughout the system. For stereo depth estimation, we employ the Neural Depth Engine developed by StereoLabs, which integrates seamlessly with the AGX GPU and provides native support for the ZED X camera platform.

The wire detection module identifies individual wire instances in each frame and estimates their corresponding depths. This stage fuses two dimensional image cues with three dimensional geometric information, which are further refined through filtering and robust outlier rejection to suppress noise and produce stable wire estimates.

The wire tracking module then associates wire instances across consecutive frames. It leverages geometric and temporal cues to match new detections with previously tracked wires and predicts their expected appearance in future frames to maintain continuity. By preserving a locally consistent representation of the wire environment, this component provides stable and reliable wire state estimates over time.

Finally, the system uses the output of the wire tracking module to select a specific wire for approach. A point on the target wire is identified in the camera frame, and

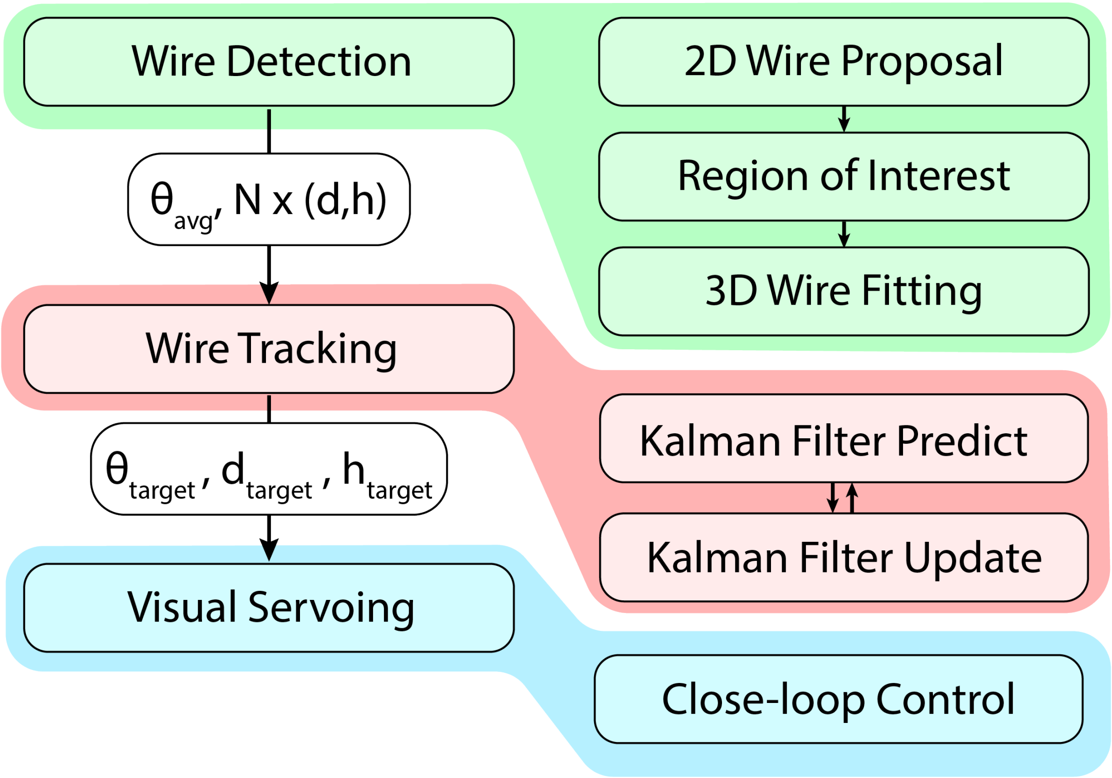


Figure 3.1: Wire approach software system diagram

the error between this desired target point and the current drone position is computed. A position based visual servoing controller then applies a PID loop to drive this error toward zero, commanding the drone to maneuver until it is aligned with and properly positioned relative to the selected wire. This closes the perception to action loop and enables controlled, stable under wire approach.

Figure 3.1 presents a system level overview of this perception pipeline.

3.2 Wire Detection

The wire detection pipeline operates on each frame of the incoming video stream and is divided into three main stages, outlined in Fig. 3.2. The first stage produces an initial estimate of the two dimensional locations of wires in the image using the RGB data. It outputs potential wire candidates, each represented by an approximate

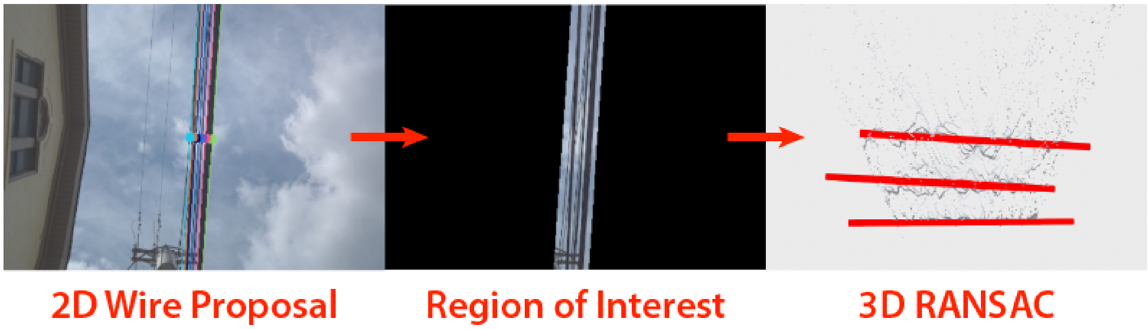


Figure 3.2: Wire instance detection pipeline

center point and an estimated average direction. Because this stage can generate false positives, the second stage incorporates geometric wire characteristics to determine regions of interest where wires are most likely to appear. This filters out extraneous responses from the two dimensional image by exploiting the characteristic geometric signature that wires produce in the depth map. By jointly analyzing the RGB and depth images, the system identifies candidate regions that can be lifted into three dimensional space, along with an estimate of how many wires are present in each region. In the final stage, these regions are projected into a point cloud, where three dimensional line fitting is applied to obtain a robust estimate of each wire's position and orientation. This pipeline not only detects wires but also produces instance level detections that spatially differentiate individual wires within each frame.

A critical design requirement for this algorithm is real time performance, running as close as possible to the frame rate of the camera. Because visual servoing demands high frequency perception updates to provide meaningful control feedback, the algorithm is designed to be computationally lightweight and to operate at or exceed thirty frames per second.

Meeting this performance target required several system level optimizations. The image resolution was reduced to 480 by 270 pixels to ensure consistent processing speed and maintain real time operation. In addition, the message transport configuration was tuned beyond the standard settings provided in ROS 2 Humble. RTI DDS [13] was selected as the communication middleware, and together with optimized image transport settings, it enabled a steady camera throughput of approximately 22 FPS.

3.2.1 2D Wire Candidate Proposal

For the two dimensional wire instance detection step, we first apply the Canny edge detection filter [2] to the grayscale version of the left camera image to extract prominent edges. The following formulation of the Canny edge detector is used to produce the edge enhanced image from the grayscale input, where G_x and G_y denote the image intensity gradients in the x and y directions, respectively:

$$\text{Edge Gradient } (G) = |G_x| + |G_y| \tag{3.1}$$

This produces a mask that highlights edge features that, in the context of wire detection, emphasize the elongated contours characteristic of wires. The resulting edge map is then passed through a Probabilistic Hough Line Transform [10], which acts as a filter to distinguish generic edges from coherent line segments, thereby isolating candidate wire structures in the image. From these detected lines, we also extract their average direction, providing an estimate of the overall wire orientation. This approach assumes that the wires share a generally consistent yaw angle, a reasonable assumption given that most power line conductors run parallel to one another.

Once the average direction has been determined, each detected line is decomposed into its constituent pixels, which are then projected onto an axis perpendicular to the average wire direction. This produces a one dimensional distribution of pixel locations along the perpendicular axis, effectively casting votes for the most likely wire edge positions. After accumulating these votes, the distribution is normalized and a peak detection algorithm is applied to identify the most probable wire edge locations. Peaks below a given threshold are discarded to ensure that short or noisy edge fragments are not incorrectly classified as wires. This process is illustrated in Figure 3.3.

The output of this stage consists of two dimensional wire candidates, each with an estimated average direction and an estimated distance from the image center measured along the axis perpendicular to the wire orientation. The full pipeline is illustrated in Fig. 3.4. Depending on the distance to the wire, the detector may observe either two edges per wire at close range or only a single edge at larger distances. Consequently, additional filtering is required to correctly attribute depth measurements to each wire candidate.

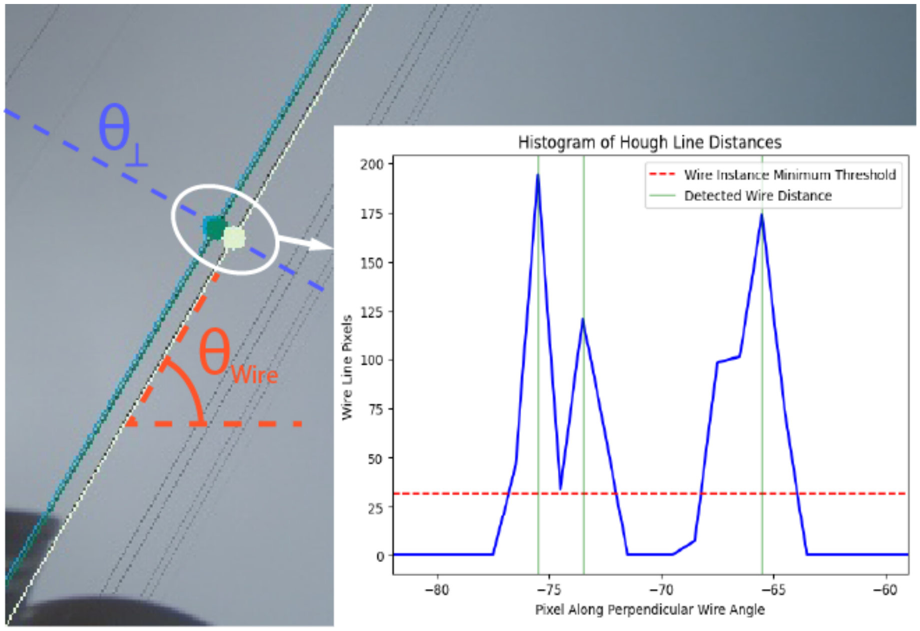


Figure 3.3: Hough Line wire instance detection along the perpendicular wire angle using peak detection

3.2.2 Region of Interest Filtering

Although the two dimensional wire candidate proposals provide an initial estimate of where long linear edges appear in the image, they rely solely on RGB information and do not verify whether these edges are geometrically consistent with a physically plausible wire projected into three dimensional space. As a result, the initial proposals highlight potential edge locations but cannot confirm whether they correspond to actual wires.

To incorporate depth information, the algorithm analyzes patterns consistent with the expected depth profile of wires, which typically exhibit a steep positive gradient followed by a negative gradient in the direction perpendicular to the wire orientation. By computing the first and second derivatives of the depth signal along this perpendicular axis, the system identifies points where the gradient undergoes a sharp change, detecting regions where wires are likely to appear.

Prior work [7] estimates wire centers by locating zero crossings in the gradient, effectively finding the midpoint between positive and negative slopes. However, this approach is prone to error: the gradient is often asymmetric on either side of the

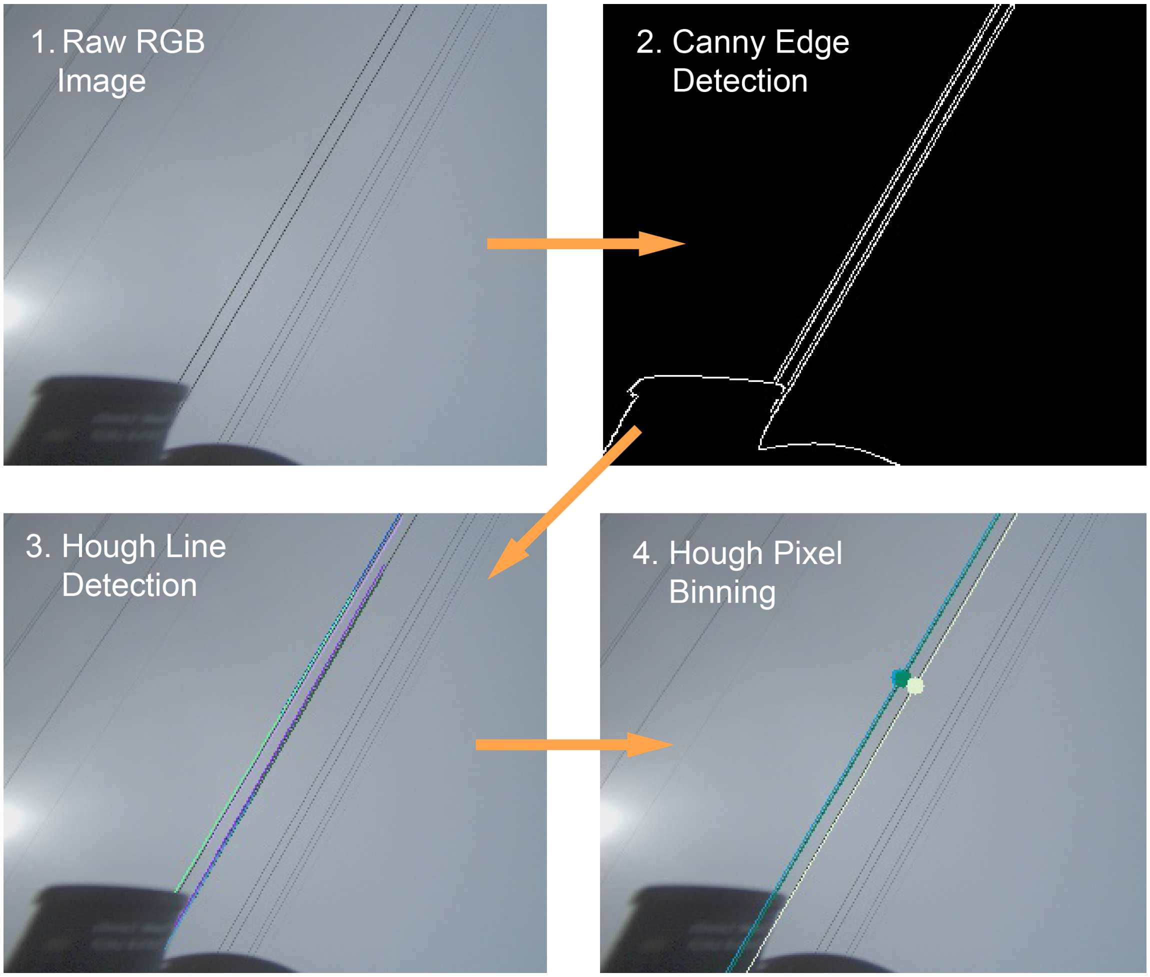


Figure 3.4: 2D wire candidate proposal process done in the RGB space

wire, and when multiple wires are close together, smoothing effects from neural depth estimation blend their profiles. This blending prevents the formation of distinct gradient centers and introduces ambiguity in locating individual wires. Consequently, we use these gradients primarily to define regions of interest, allowing more precise wire localization to be achieved through point fitting in subsequent steps.

$$DepthGradient(G) = G_x \cos(\theta_\perp) + G_y \sin(\theta_\perp) \tag{3.2}$$

In our method, we first compute the Sobel gradients in the x and y directions and combine them to calculate the gradient along the axis perpendicular to the wire, as described in Section 3.2.1 and shown in Equation 3.2. We then project the depth gradients onto this perpendicular axis, following the same approach.

After projection, each pixel is represented by two quantities: its signed distance along the perpendicular axis from the center of the image, and its corresponding depth gradient value. The distance axis is discretized into integer bins, and the gradient values within each bin are aggregated. For each bin, the total gradient magnitude is divided by the number of contributing pixels to produce an average gradient profile along the perpendicular sweep. The normalized absolute gradient is then differentiated to detect transitions where the slope changes significantly. These zero crossings mark the beginning and end of potential regions of interest. Starting from the leftmost part of the image, adjacent regions are merged to form continuous intervals where wires are likely to be present.

To eliminate false regions of interest, we retain only those ROIs that contain at least one two dimensional wire candidate, ensuring that the remaining areas are supported by both visual evidence and geometric depth cues. This modality combination process is illustrated in Fig. 3.5. Once identified, these regions are overlaid onto the depth image to produce segmented point clouds corresponding to likely wire locations. Each region also carries an estimate of the number of wire instances it contains, which is essential for constraining the subsequent three dimensional fitting process. Since three dimensional fitting is computationally intensive, reducing the search space and limiting the number of points per region is critical for real time performance. This filtering stage therefore plays a key role in enabling the algorithm to operate at the desired frame rate.

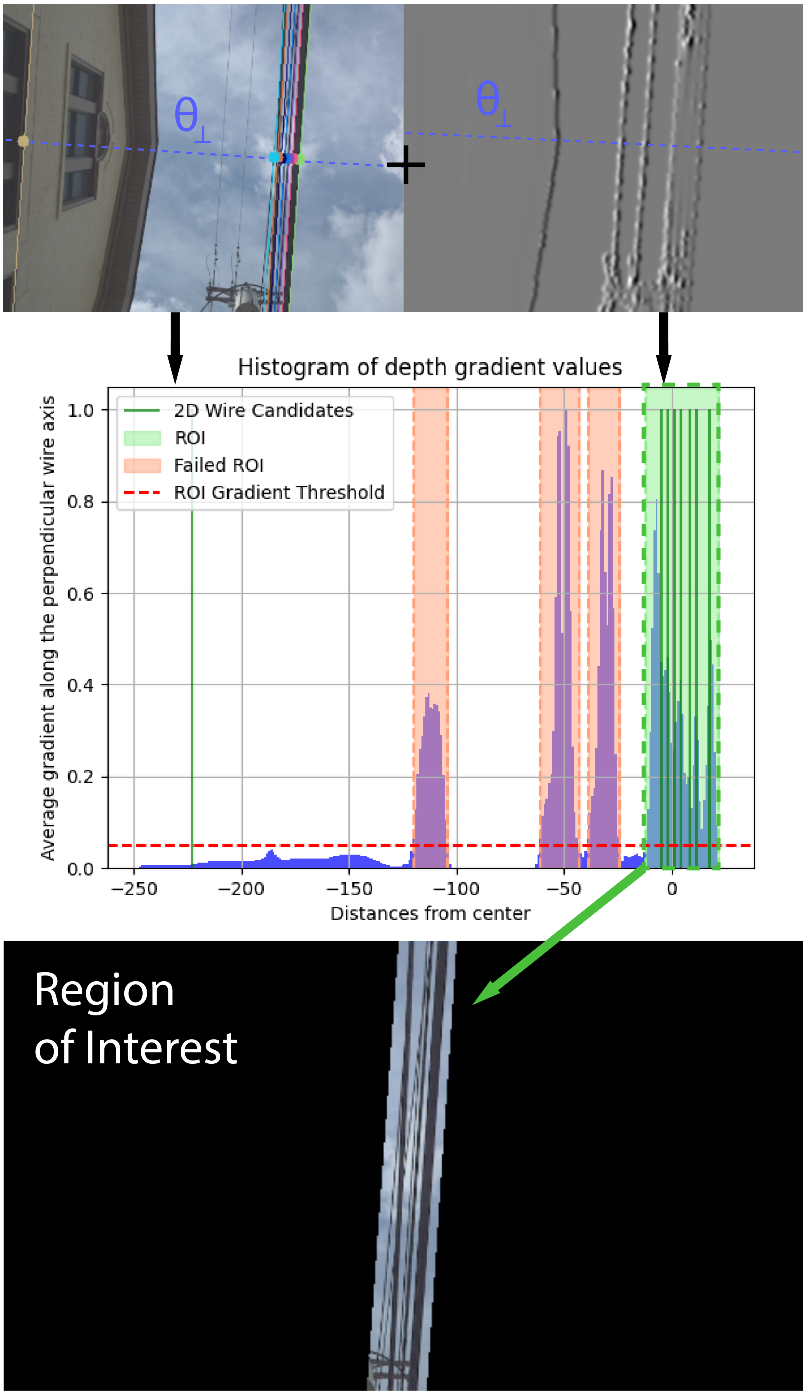


Figure 3.5: Wire region of interest mask creation

3.2.3 3D Wire Fitting

Once the regions of the depth and RGB images have been identified and the expected number of wire instances within each region has been estimated, the next step is to perform robust line fitting. This step is necessary because the point clouds generated from dense depth estimation are highly noisy along the length of the wire. Due to the characteristics of neural dense stereo and the textureless appearance of wires, the depth estimates exhibit significant variability. Moreover, these variations are not well modeled by a Gaussian distribution; instead, the predicted wire often appears wavy or irregular in the reconstructed point cloud. As a result, extracting the true underlying line structure is challenging, and any fitting method must be robust to significant outliers.

To address this, our method employs a parameter-constrained RANSAC procedure [3]. Using information from the two dimensional wire proposals and the regions of interest, we restrict the scope of RANSAC sampling to enable real time execution. The point cloud is first constrained by enforcing minimum and maximum depth bounds. A mask is also applied to remove any portions of the UAV visible in the camera frame, preventing non-wire points from being incorporated into the fitting process. Additionally, the fusion of the ROI and two dimensional wire proposals spatially limits the point cloud in the x and y directions, producing compact regions where line inference is performed.

To fit a 3D line model, pairs of points are randomly sampled from the point cloud and first verified against a set of constraints before computing inliers, which is the most computationally intensive step. Table 3.1 summarizes these constraints, which include spatial limits, angular limits, and confidence criteria. The angular constraints enforce consistency with the expected geometry of power lines, which are generally level and aligned with the yaw estimated from the 2D detection in Section 3.2.1. The confidence constraints ensure that a candidate line has sufficient support from nearby points of adequate quality. Together, these constraints substantially reduce the number of implausible proposals and allow RANSAC to operate efficiently while remaining robust to noise. A visualization of the RANSAC line fittings is seen in Figure 3.6.

Ultimately, this step provides the wires' angles, their heights, and their distances

Constraint	**Margins**	**Details**
Depth Minimum	≥ 0.5 meters	Minimum depth points are considered
Depth Maximum	< 10.0 meters	Maximum depth points are considered
Wire Vertical Direction	± 10 degrees	Maximum angle deviation from a level line
Wire Horizontal Direction	± 4 degrees	Maximum angle deviation from detected wire angle
Wire Count per ROI	$< \max\left(\frac{\text{ROI Line Count}}{2}, 1\right)$	Maximum number of wire instances in a ROI
Inlier threshold	5 inches	Distance from a point to a line to be cconsidered in its confidence
Minimum Point threshold	5%	Minimum percentage of inliers versus point cloud points to be considered valid
Fitting Iterations	600 count	Maximum amount of time RANSAC can query

Table 3.1: RANSAC constraints applied to line fitting

from the center along the direction perpendicular to the wires in the frame. These measurements can then be used in Section 3.3 of the algorithm to track individual wire instances across frames.

3.2.4 Wire Detection Model

Through extensive collection of under-wire imagery and the development of a novel multi-wire detection algorithm, we can automatically generate a large dataset of labeled wire detections. This enables training a robust detection model that automates the process and improves resilience to noise. Since each data run provides both 3D wire line estimates and corresponding camera images, we are able to then project the 3D wire geometry into the image frame to generate 2D bounding boxes, as shown in Figure 3.7. Appropriate bounding box dimensions are estimated based on the 3D geometric properties of the wire, with a small padding applied to produce high-quality

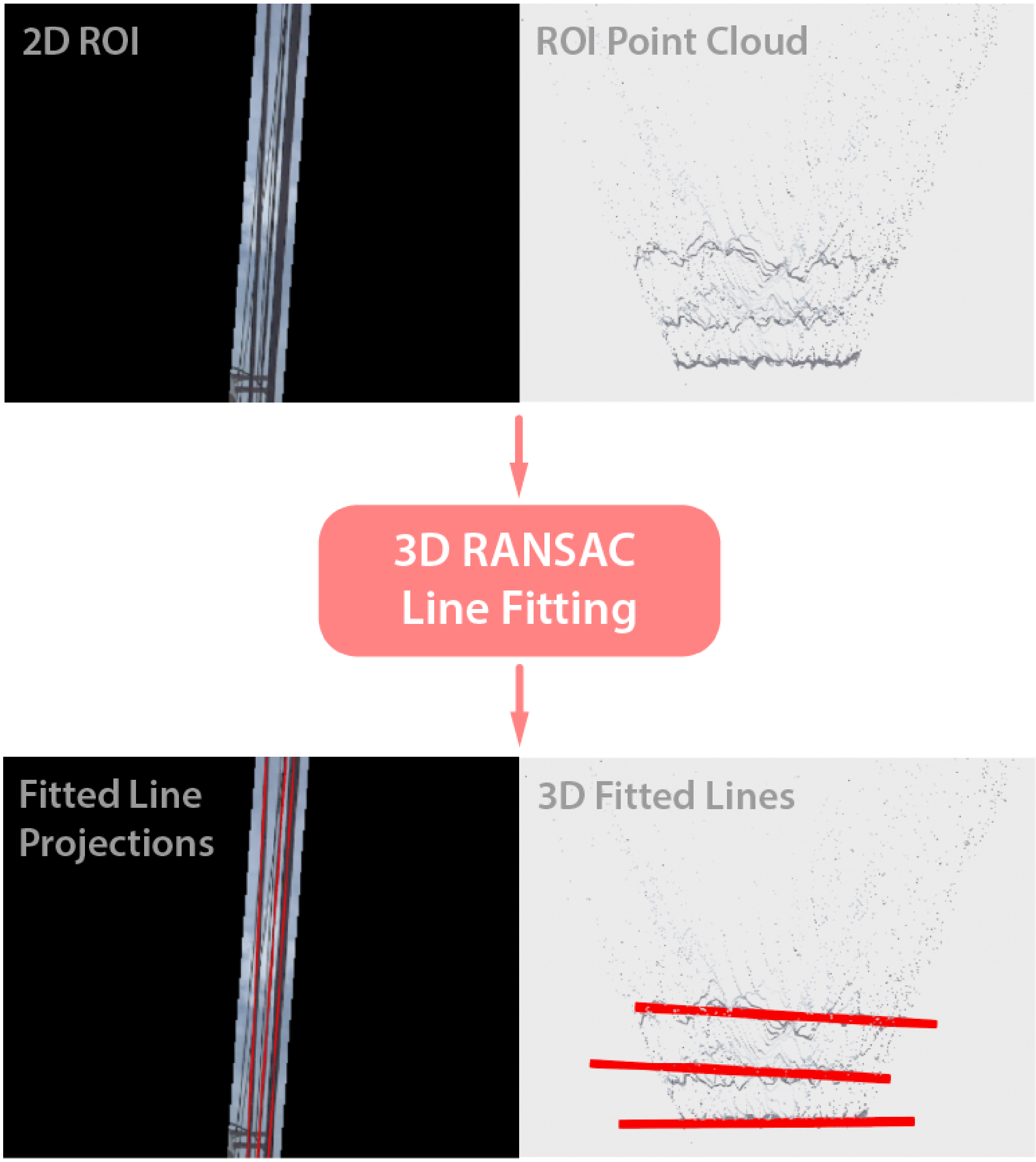


Figure 3.6: RANSAC wire fitting output for a single ROI point cloud

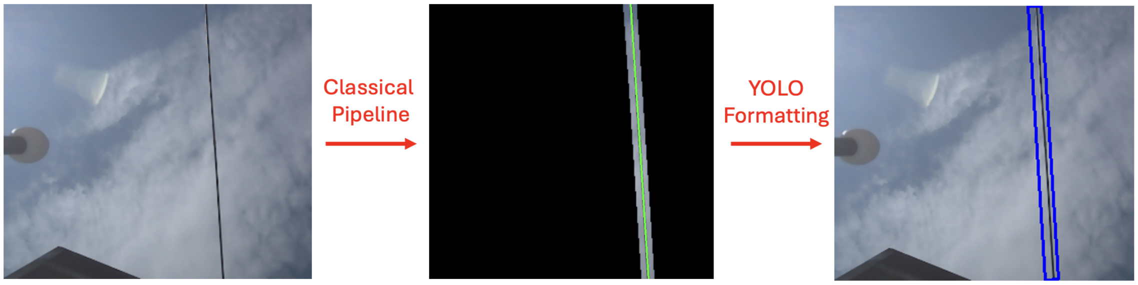


Figure 3.7: Generation of ground truth bounding boxes using the classical wire detection pipeline

labels. Equation 3.3 demonstrates how the width and depth of the wire are projected into the image frame to determine the size of the bounding box.

$$BB_{width} = \frac{f_{cam} W_{wire}}{Z_{wire}} + padding \tag{3.3}$$

Two primary requirements guided the selection of the model architecture to be able to fit into our pipeline. First, the model must operate in near real time, as closed-loop visual servoing demands rapid responsiveness to vehicle motion for stable control and timely feedback. Second, the model must produce oriented bounding boxes aligned with the wire's geometry. Such alignment reduces the search space for downstream 3D line fitting and constrains the RANSAC procedure. To preserve real-time performance, the bounding boxes are kept as tight as possible to prevent nearby structures or access points from contaminating the RANSAC inference. Considering these requirements, we selected the YOLO11 Oriented Bounding Box variant [6] as the base detection model.

The model is initially pretrained on the DOTA dataset, which contains oriented bounding boxes derived from aerial imagery. Only the cleanest detection runs processed through the classical wire detection pipeline were incorporated into the training dataset, with manual review used to remove a small number of misdetections and ensure label quality. Extensive data augmentation was applied to increase dataset variability, with particular emphasis on orientation, translation, and scale transformations to maximize geometric diversity. Color augmentations, especially variations in saturation, were also applied to better replicate the overcast sky conditions commonly encountered during wire tracking experiments. Figure 3.8 demonstrates different

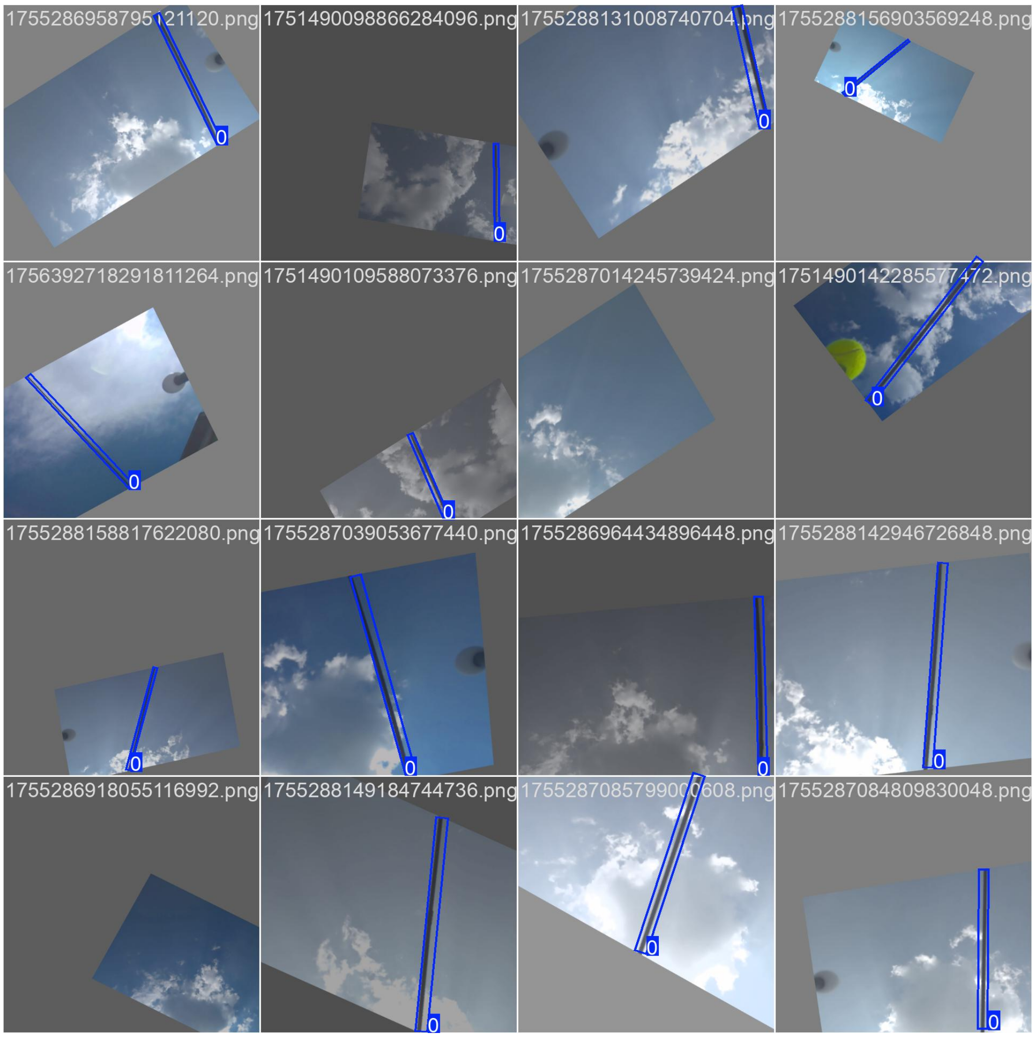


Figure 3.8: A single batch of wire detection training images with diverse augmentations

types of augmentations done on the training data. In total there was 11,212 images used in the testing and validation of the model.

3.3 Wire Tracking

While the wire detection module provides per-frame wire locations, it does not ensure temporal consistency. To allow the UAV to lock onto a specific wire and generate motion commands, we need to track individual wire instances across frames. This is challenging because wires lack distinctive visual features, appear thin at longer ranges, and can cluster together, making it difficult to reliably associate detections over time.

Our method primarily relies on position-based correlation between consecutive detections to overcome this challenge. However, position-based tracking is complicated by UAV pose drift, as onboard sensors and GPS provide noisy, meter-scale estimates. Therefore, the tracking algorithm must tolerate imperfect state estimates while maintaining local consistency.

To address this, the system uses Kalman filtering to manage uncertainty in wire positions. It employs two complementary filters: an orientation filter that tracks the shared wire direction and per-instance position filters that estimate each wire's depth and perpendicular offset in the camera frame. The orientation filter processes the mean wire direction as a unit vector, averaging detection angles and restricting the vector to a single hemisphere to prevent flips.

Position filters maintain a full covariance matrix and a valid count to track observation frequency. Raw image measurements are converted to 3D coordinates and then reprojected into depth and perpendicular distance, so these filters are implemented as Extended Kalman Filters. This approach incorporates new detections while removing stale or inconsistent tracks representing outliers. Figure 3.9 illustrates the Kalman Filter configuration.

A central design goal of the tracking framework is robustness to abrupt jumps or drift in the UAV's global pose. This is achieved by representing wire positions using relative pose transformations between consecutive frames, rather than relying on absolute global poses. Using relative motion prevents a single corrupted pose estimate from invalidating all wire tracks, allowing the Kalman filters to absorb residual errors and maintain stability.

To compute the relative transform between flight controller poses in the camera frame, let $H_{\text{cam}}^{\text{fc}}$ denote the homogeneous transform from the camera to the flight controller, $H_{\text{cam}}^{\text{world}}$ the transform from the camera to the world, and $H_{\text{fc}}^{\text{world}}$ the transform from the flight controller to the world. The flight controller provides $H_{\text{fc}}^{\text{world}}$ at each time step. Using a prime symbol ($'$) to denote the future pose, the relative transform in the camera frame is then given by Equation 3.4.

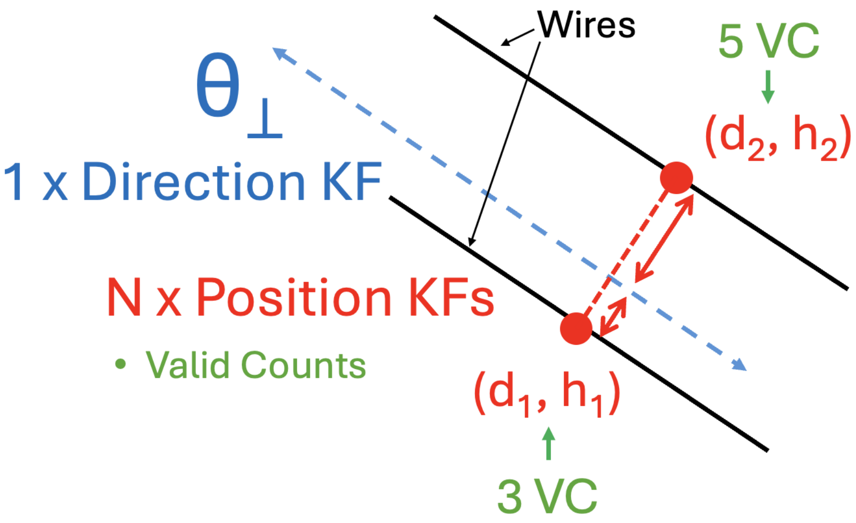


Figure 3.9: The Kalman filter paradigm for wire tracking, using a single direction filter and a per-instance set of position filters

$$\begin{aligned} H^{world}_{cam} &= H^{world}_{fc}\, H^{fc}_{cam} \\ H^{cam'}_{cam} &= \left(H^{world}_{cam'}\right)^{-1} H^{world}_{cam} \end{aligned} \tag{3.4}$$

In Figure 3.10, the frame conventions used throughout the system are illustrated on the VTOL platform. Using the relative transform from the previous camera pose to the current camera pose, we update the position and orientation Kalman filters to reflect the newly expected wire environment. These updated predictions are then matched against the wire instances detected in the current frame.

The direction Kalman filter estimates the orientation of the wire using a unit vector that represents both its horizontal yaw and vertical pitch. The prediction step for this direction vector is given Equation 3.5.

$$\begin{aligned} X' &= R^{cam'}_{cam}\, X \\ P' &= R^{cam'}_{cam}\, P\, (R^{cam'}_{cam})^T + Q \end{aligned} \tag{3.5}$$

To perform the update step of the direction Kalman filter, we consider all fitted wire lines in the current frame and compute their directional mean to obtain a unit vector that represents the estimated wire orientation. The direction estimate in the Kalman filter is then updated using Equation 3.6.

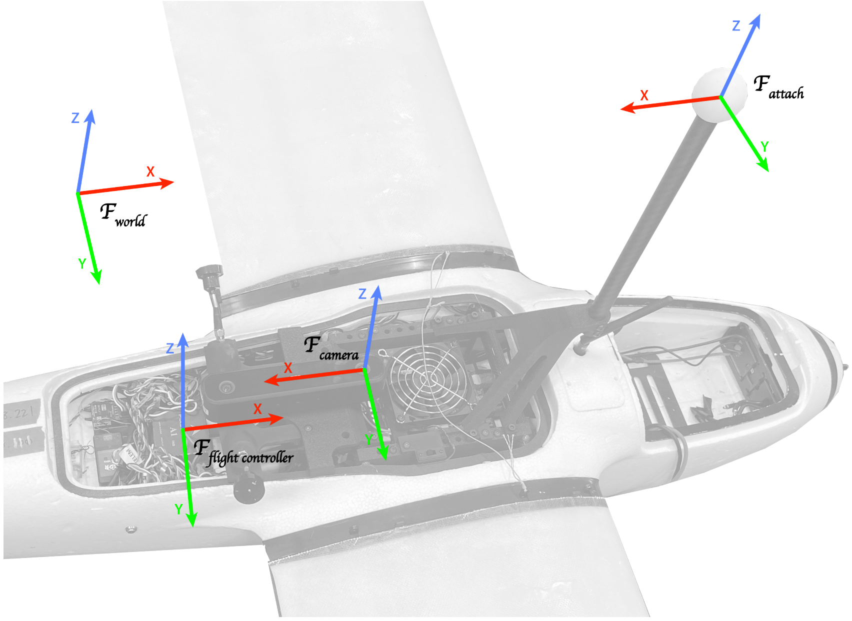


Figure 3.10: VTOL frame conventions

$$\begin{aligned} S &= P' + R \\ K &= P'S^{-1} \\ Y &= X_{\text{measured}} - X' \\ X'' &= \|X' + KY\| \\ P'' &= (I - K)P' \end{aligned} \tag{3.6}$$

For position tracking, the position is represented as the distance from the wire center along the direction perpendicular to the wire orientation, as well as the height relative to the drone. An Extended Kalman Filter (EKF) is employed because the measurements must first be converted to Cartesian coordinates for transformation, and then converted back to the marginalized position representation. The prediction step is performed for each tracked wire individually. Specifically, the prediction is computed using Equation 3.7.

$$
\begin{aligned}
X_{dh} &= \begin{bmatrix} d \\ h \end{bmatrix} \\
X_{xyz} &= \begin{bmatrix} -\sin(\theta_\perp) & 0 \\ \cos(\theta_\perp) & 0 \\ 0 & 1 \end{bmatrix} X_{dh} \\
X'_{xyz} &= RX_{xyz} + t \\
X'_{dh} &= \begin{bmatrix} -\sin\theta'_\perp & \cos\theta'_\perp & 0 \\ 0 & 0 & 1 \end{bmatrix} X'_{xyz} \\
J &= \begin{bmatrix} \begin{matrix} \sin\theta'_\perp r_{11}\sin\theta_\perp - \sin\theta'_\perp r_{12}\cos\theta_\perp \\ -\cos\theta'_\perp r_{21}\sin\theta_\perp + \cos\theta'_\perp r_{22}\cos\theta_\perp \end{matrix} & \cos\theta'_\perp r_{23} - \sin\theta'_\perp r_{13} \\ -\sin\theta_\perp r_{31} + \cos\theta_\perp r_{32} & r_{33} \end{bmatrix} \\
P' &= JPJ^T + Q
\end{aligned}
\tag{3.7}
$$

Once the wire's predicted distance and height are computed based on the updated perpendicular yaw angle, the update step incorporates the newly observed wire positions into the Kalman filter for each tracked wire. Association is performed geometrically: in-frame wire observations are paired with existing tracks if they fall within a distance threshold, and new tracks are initialized for observations not near any existing wire. For each matched wire, the update step is performed using Equation 3.8.

$$
\begin{aligned}
S &= P' + R \\
K &= P'S^{-1} \\
Y &= X_{\text{measured}} - X'_{dh} \\
X'' &= X' + KY \\
P'' &= (I - K)P'
\end{aligned}
\tag{3.8}
$$

The position update step also includes logic for managing the addition of new wire instances and the removal of inaccurate tracks resulting from noise. This is handled using a validity count for each wire Kalman filter. When a wire track is initialized, it starts with a default valid count to prevent immediate removal. In subsequent frames, if an observation is associated with a tracked wire, its valid count is incremented. Conversely, if a wire is expected to appear in the camera frame but no corresponding observation is detected within the distance threshold, the valid count is decremented. Any wire whose valid count reaches zero is removed from tracking. If a wire is observed that does not correspond to an existing track, a new Kalman filter is created to track it. The pseudo code in Algorithm 1 illustrates the logic of the position update function, outlining the decision-making process based on the association of existing Kalman filters with newly observed wires.

In summary, the proposed wire tracking pipeline robustly tracks individual wire instances over time. By integrating Extended Kalman Filters, geometric association, and validity-based management, it maintains continuous tracks while reducing noise and inaccurate detections. This framework enables reliable wire localization in dynamic scenes and supports the selection of an optimal wire target for downstream tasks.

3.4 Position-Based Visual Servoing

Building on the wire detection and tracking systems, we leverage instance-level tracking of each individual wire to select a target for approach. A wire is considered a valid candidate once its tracking validity count exceeds a predefined threshold. Among these valid wires, the one closest to the drone in the local frame is selected as the active target to avoid occlusion or interference from other wires. To define the primary target point on this wire, we use the estimated wire distance and height to parameterize it as a 3D line relative to the camera frame. These quantities are then expanded along the camera's x, y, and z axes to obtain a full Cartesian target position, as shown in Equation 3.9.

$$\begin{bmatrix} x_{target} \\ y_{target} \\ z_{target} \end{bmatrix} = \begin{bmatrix} -\sin(\theta) * d \\ \cos(\theta) * d \\ h \end{bmatrix} \tag{3.9}$$

To approach the target reliably, we employ a position based visual servoing (PBVS)

Algorithm 1 Updating Position Kalman Filters with Wire Instance Estimates

```
function updatePositionKalmanFilters($observed\_wires \in \mathbb{R}^{3\times N}$, $\theta_{\perp}$)
    $dhs\_observed \leftarrow$ getDHFromXYZs($observed\_wires, \theta_{\perp}$)
    $distances\_matrix \leftarrow$ getDistanceMatrix($dhs\_observed, dhs\_kfs$)    ▷ $N \times M$ matrix
    $distances\_masked \leftarrow (distances\_matrix < wire\_matching\_min\_threshold)$
    $kfs\_matched \leftarrow$ columnwise_OR($distances\_masked$)    ▷ $1 \times M$ vector
    $observed\_matched \leftarrow$ rowwise_OR($distances\_masked$)    ▷ $1 \times N$ vector
    for each $i$ in length($kfs\_matched$) do
        if $kfs\_matched[i] == 1$) then
            $kf[i]$.incrementValidCount()
        else
            $in\_frame \leftarrow$ checkKFInFrame($kfs[i]$)
            if $in\_frame$ then
                $kfs[i]$.decrementValidCount()
                if $kfs[i]$validCount $\leq 0$ then
                    $kfs$.removeStaleKF($i$)
                end if
            end if
        end if
    end for
    for each $i$ in length($observed\_matched$) do
        if $observed\_matched[i] == 0$) then
            $kfs.addKF(dhs\_observed[i])$
        end if
    end for
end function
```

strategy, which uses visual information to regulate robot motion by extracting features from images and computing a control law. In PBVS, the full relative pose between the camera and the target is reconstructed, allowing the control error to be defined in Cartesian space rather than in image coordinates. This is particularly suitable for our method, since the tracking system provides metrically meaningful estimates of the wire's position and orientation, enabling the controller to treat the wire as a spatial reference.

The perception module continuously outputs the relative pose of the selected wire in the drone's body frame, including lateral, vertical, and forward offsets as well as relative yaw. These quantities form the PBVS error vector, representing the deviation between the drone's current configuration and the desired pose in which the wire is centered and aligned in front of the vehicle. PBVS aims to regulate this error vector to zero.

To achieve this, we apply a proportional–integral–derivative (PID) control law independently to each component of the pose error. The PID controller combines instantaneous error feedback, accumulated bias correction, and predictive damping to provide robustness against estimation noise, modeling inaccuracies, and small disturbances. The resulting control law maps the pose error into body-frame velocity commands, as expressed in Equation 3.10:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\theta} \end{bmatrix} = K_p \begin{bmatrix} e_x(\tau) \\ e_y(\tau) \\ e_z(\tau) \\ e_\theta(\tau) \end{bmatrix} + K_i \int_0^t \begin{bmatrix} e_x(\tau) \\ e_y(\tau) \\ e_z(\tau) \\ e_\theta(\tau) \end{bmatrix} d\tau + K_d \frac{d}{dt} \begin{bmatrix} e_x(\tau) \\ e_y(\tau) \\ e_z(\tau) \\ e_\theta(\tau) \end{bmatrix} \tag{3.10}$$

These PID loops convert positional and angular errors into real-time body-frame velocity commands: lateral errors produce sideways motion, vertical errors adjust height, forward errors regulate distance, and yaw errors align the drone with the wire. As the drone moves, the perception system updates the wire pose, reducing the error vector and allowing the controller to bring the vehicle smoothly into stable alignment, as illustrated in Fig. 3.11.

To maintain a safe interaction zone, positional offsets are incorporated into the target location so the drone approaches an ideal vantage point rather than the wire itself. This closed-loop feedback structure ensures robust convergence even in the

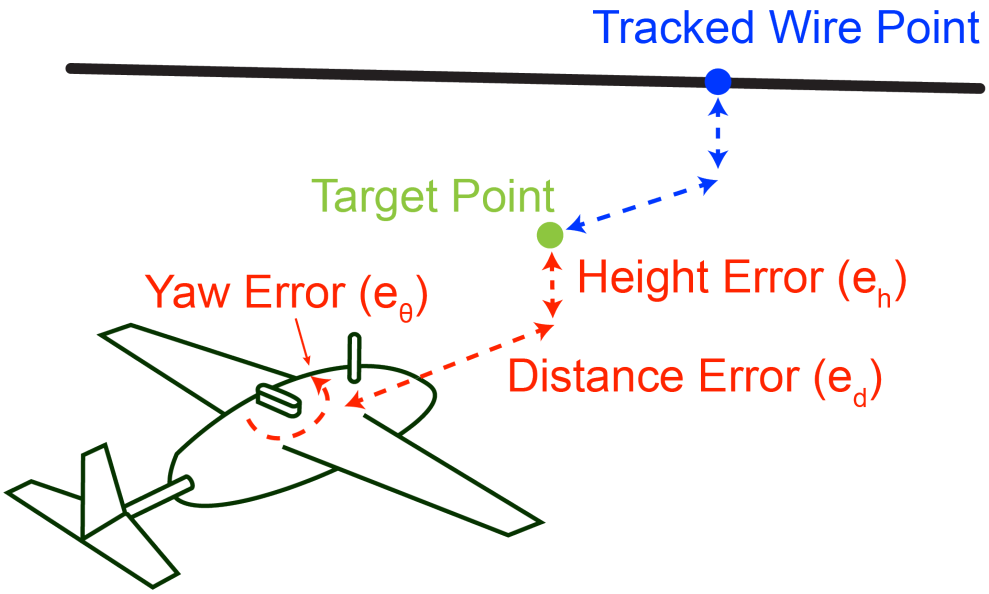


Figure 3.11: Position-based visual servoing scheme

presence of noise, partial occlusions, or small disturbances, enabling reliable approach and alignment with thin, visually sparse wires.

3.5 Hardware System

For our hardware, we required a system that would integrate seamlessly with a stereo depth camera while remaining modular enough to be deployed across multiple flight platforms. This necessitated selecting a modular compute system and peripherals capable of supporting communication, remote control, and position estimation. A full layout of our hardware system is shown in Figure 3.12.

To meet these requirements, we used an NVIDIA Jetson Orin AGX 64GB, providing a centralized, high-performance platform capable of supporting wire approach autonomy as well as additional mission-specific tasks to extend operational range. A battery elimination circuit (BEC) was employed to step down battery voltages, ranging from 3S to 12S, to a stable 12V supply, delivering sufficient current to operate the Orin AGX at its maximum power mode.

For the flight controller, we required a framework that would be generalizable

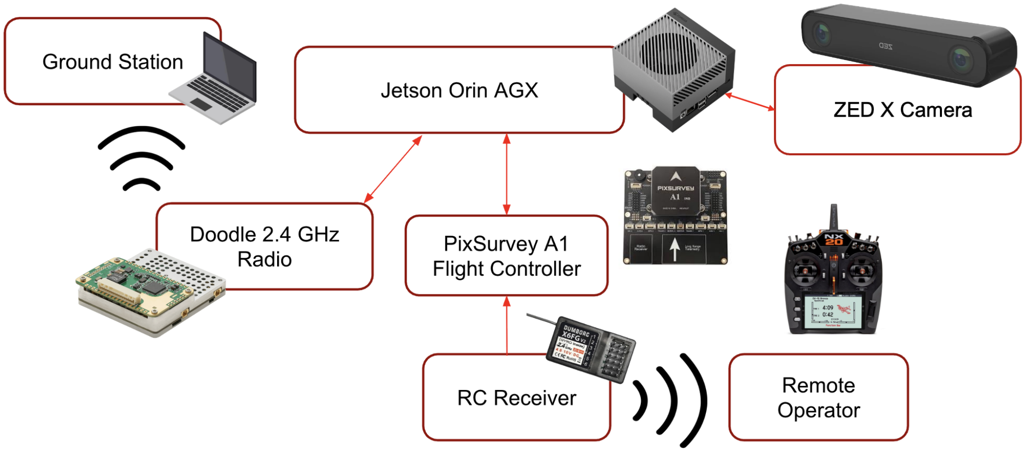


Figure 3.12: Wire approach hardware system diagram

across different drones while supporting teleoperation, autonomous functionality, and position estimation. We selected flight platforms running ArduPilot firmware to meet these requirements. Communication between the onboard computer and the flight controller was established via MAVROS over a micro-USB connection, enabling the onboard system to receive platform pose estimates and send velocity-based control commands directly from the wire approach autonomy stack to govern the drone's motion.

For high-bandwidth communication of wire estimates, a Doodle Labs 2.4GHz Mesh Rider Radio was installed. This radio provided reliable data transmission between the UAS and the ground control station and supported real-time video streaming during flight tests. It was connected to the Orin AGX via Ethernet for bidirectional communication and powered through a 12V-to-5V converter tapped from the flight controller's power bus.

To complement the supporting hardware architecture, it was necessary to select a stereo camera that would meet our requirements and provide accurate depth estimates. The ZED-X camera, equipped with a 4.0mm focal length lens, was mounted facing upward and dedicated to wire tracking. Its narrower field of view improved depth perception, which is critical for precise wire localization. Experimental comparisons showed that the longer focal length lens produced more reliable depth estimates and reduced noise in the wire detection pipeline, as illustrated in Figure 3.13. Additionally, the global shutter of the ZED-X ensures accurate imaging on fast-moving UAVs,

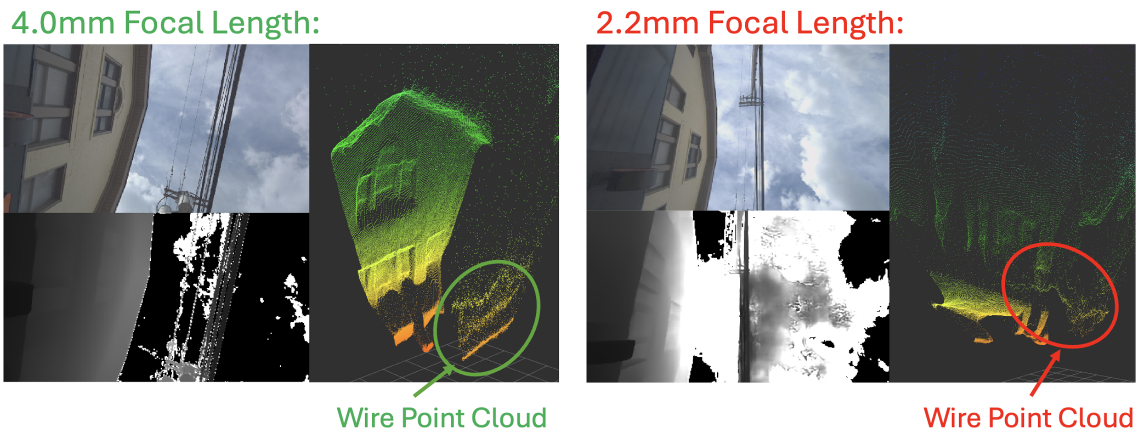


Figure 3.13: Camera focal length wire depth comparison

avoiding the distortions caused by rolling shutters that can introduce waviness in power lines, as noted in prior work [9]. Depth information is obtained using the ZED-X's neural-based depth estimation, which generates dense, full-frame depth maps and leverages the Orin AGX GPU for real-time inference.

The system was designed to be highly modular so it could be integrated into a variety of flight platforms. Figure 3.14 demonstrates its integration on one UAV platform, and the architecture was also successfully deployed on an additional platform during data collection and testing.

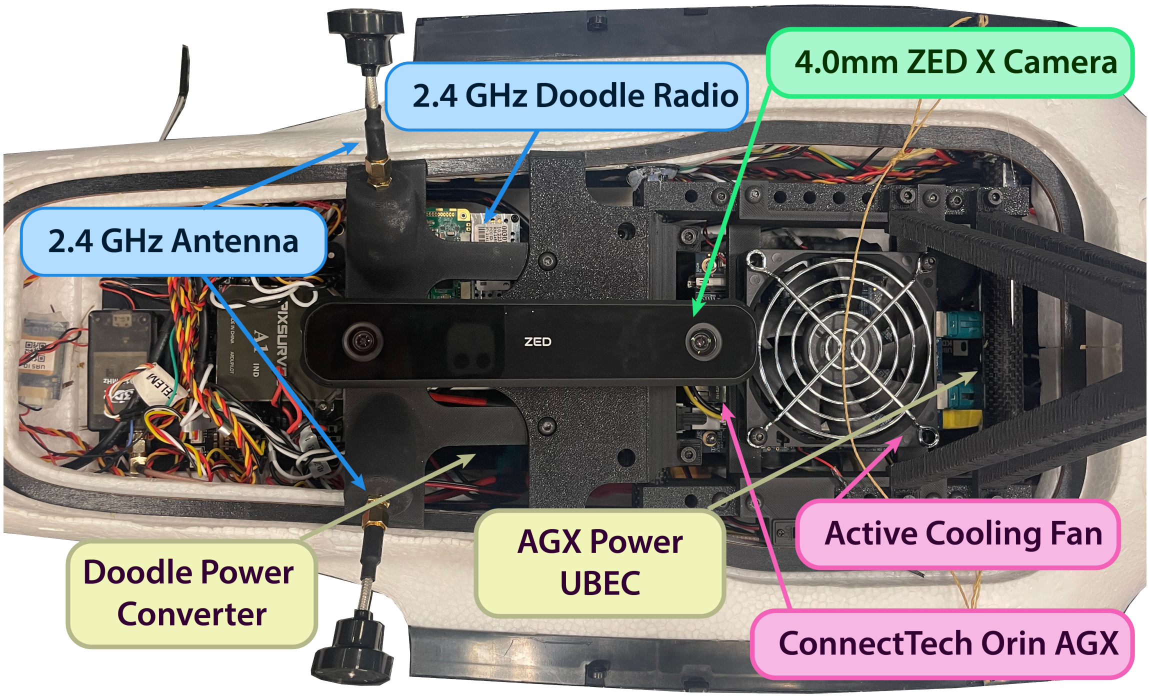


Figure 3.14: Hardware system integrated on a flight platform

Chapter 4

Experimental Setups

Testing of wire detection, wire tracking, and wire approach was conducted in three main stages across various platforms. We began in simulation to validate the algorithms and test extended functions like wire approach without risk of hardware damage. Next, we moved to controlled indoor tests and handheld outdoor experiments to evaluate real world noise, which significantly affected performance. This prompted an iterative process of refining the system in real world conditions and re-evaluating it in simulation to ensure robustness. We then collected outdoor data with different drone platforms for offline evaluation and runtime testing on edge compute hardware. Finally, the system was tested outdoors on the full platform for final evaluation.

4.1 Simulation

To evaluate our method and its variations, we developed a simulation world and model using NVIDIA Isaac Sim. To create a software in the loop interface, we connected Isaac Sim to an Ascent Spirit drone Software-In-The-Loop (SITL) dynamics model backend, which enabled us to issue flight commands and receive simulated coaxial dynamics responses. This backend provided the necessary vehicle dynamics modeling, capabilities not natively supported in Isaac Sim, and allowed these computed forces and moments to drive the vehicles motion within the simulation. As a platform, Isaac Sim offers more realistic rendering compared to other environments such as Gazebo.

In parallel, we contributed to a generalizable autonomy stack called AirStack,

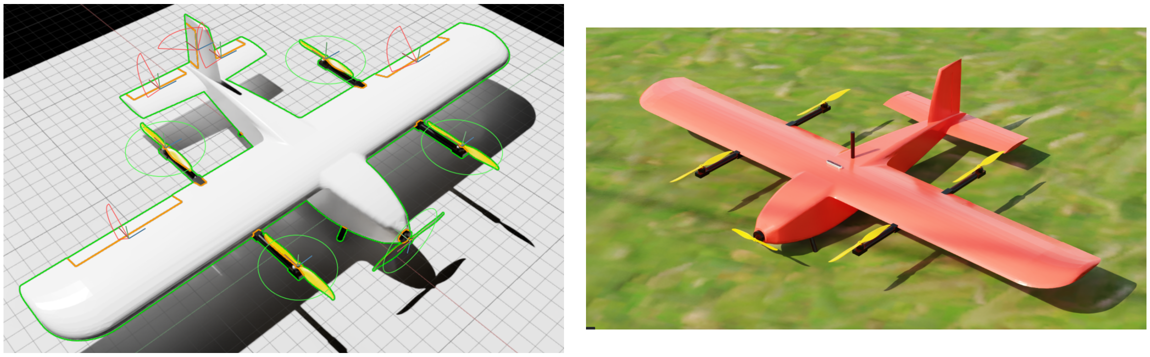


Figure 4.1: Left: Simulation VTOL control surface modeling, Right: Simulation VTOL in a test environment

which served as the backbone for our experiments. AirStack provides an interface compatible with most flight controllers and enables generalizable communication between autonomy modules and controllers across platforms. It also manages basic flight operations such as takeoff and landing and allows integration of our powerline autonomy module for enhanced functionality.

Using these frameworks, we constructed a simulation environment and software stack that closely replicate real-world operations. Our simulated world includes two power line models, shown in Figure 4.2. The first model represents power lines commonly found in densely populated urban areas, smaller in scale with multiple wires positioned closely together. The second model represents interstate power lines, which are larger and feature widely spaced wires. This variation allows evaluation of different methods across a range of realistic scenarios. Figure 4.4 shows an upward view from the VTOL's camera, highlighting the difference in wire proximity.

4.2 Indoor Setup

The controlled indoor testing environment was designed to provide a simplified and safe setting for evaluating the wire tracking algorithm. To simulate a single-wire scenario, a 1.5-inch PVC pipe was wrapped in black tape to approximate the visual appearance of power lines. A handheld jig replicated the VTOL's stereo camera configuration, including a downward-facing camera for positional reference and an upward-facing camera for wire tracking.

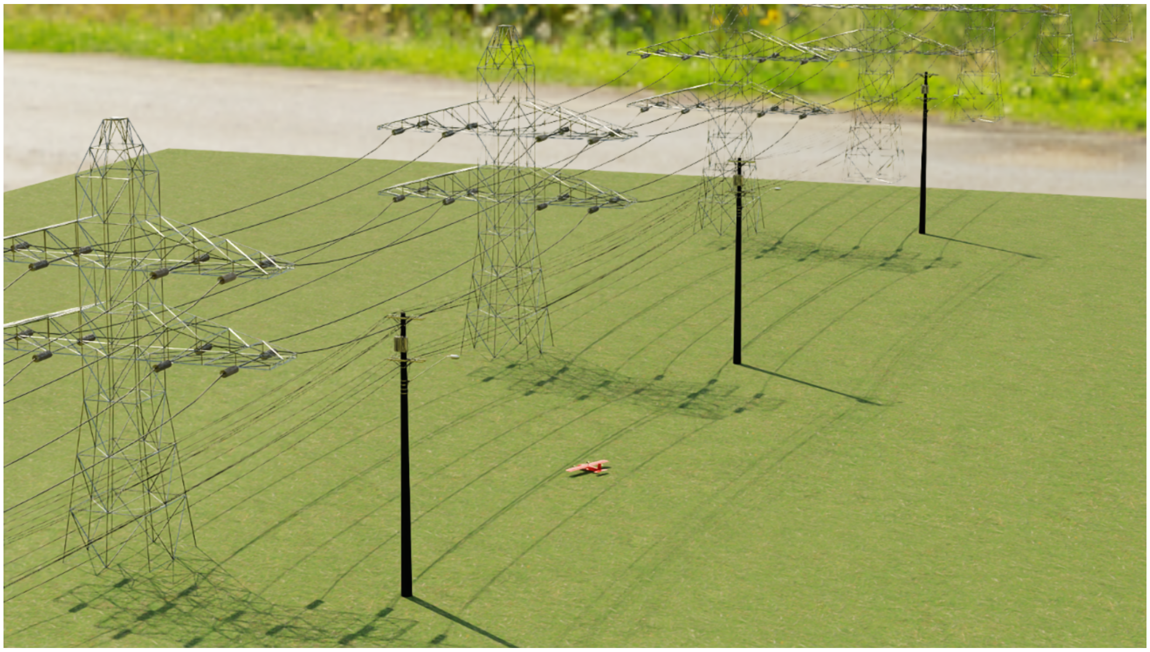


Figure 4.2: The NVIDIA Isaac Sim power line simulation environment

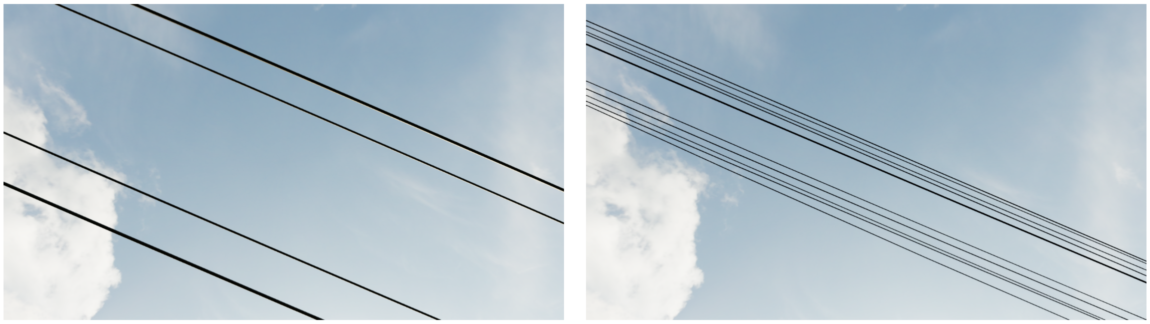


Figure 4.3: Left: Upward view from the simulated VTOL of the interstate power line model, Right: Upward view from the simulated VTOL of the urban power line model

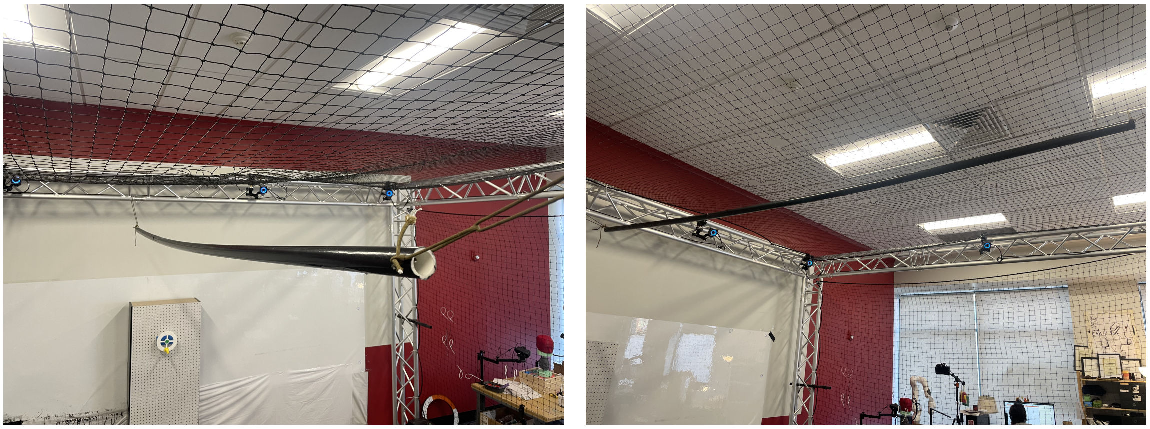


Figure 4.4: Indoor pseudo wire testing setup for controlled single wire testing

This setup allowed testing at distances up to 2.2 meters from the pseudo wire, offering flexibility in positioning while maintaining control over environmental factors such as lighting and background. It enabled systematic evaluation of the detection and tracking pipeline in a ground-based setting, without the complexity or risk of full-flight integration. Using this configuration, we efficiently conducted multiple test runs, iterated on algorithm parameters, and verified stereo depth estimation before progressing to airborne experiments. The controlled environment, combined with the ability to obtain ground truth measurements, also allowed us to establish benchmarks and assess overall system performance.

4.3 Outdoor Setup

In our outdoor testing, we evaluated several payload configurations that shared a common hardware architecture and were deployed across multiple test locations. The first payload, shown on the right side of Figure 4.6, uses a VTOL-style airframe with its control surfaces removed to reduce drag and minimize disturbances during quadrotor-mode remote control. This VTOL airframe was used for all pseudo-wire testing scenarios, as it most closely represents our target operational platform. The second payload was integrated onto the Spirit Ascent drone shown on the left of Figure 4.6, which employs a coaxial frame. This configuration featured the stereo cameras in a slightly modified arrangement while retaining the same onboard compute

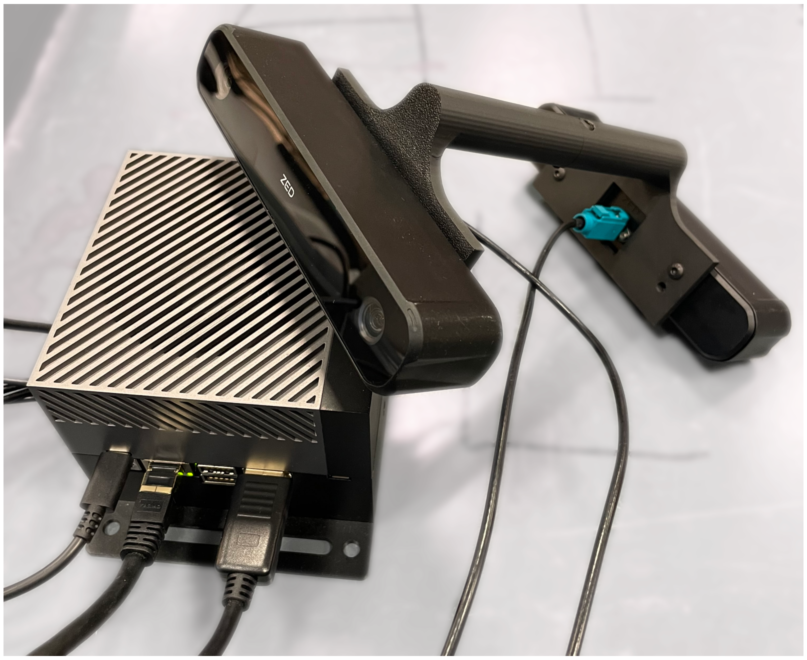


Figure 4.5: Handheld testing jig for wire tracking validation

hardware.

We conducted testing at three primary multi-wire locations and collected datasets from each, shown in Figure 4.7. These sites offered a broad range of wire configurations, providing diverse scenarios to evaluate the robustness of our pipeline. The datasets included wire heights from approximately 3 to 20 meters above ground, with variations in thickness, orientation, and background conditions. Changes in sky conditions, particularly sun glare, were found to significantly affect detection and tracking performance. These tests primarily focused on collecting data to validate the wire detection algorithm before extending to the tracking component.

In addition to real wire scenarios, we created pseudo wire environments by suspending black PVC pipes across elevated attachment points, as shown in Figure 4.8. This setup enabled safe, close-proximity flights while simulating higher-risk conditions. The pipes, approximately 1.5 inches in diameter, closely matched the thickness of typical power wires, allowing safer testing and increasing the diversity of test scenarios.

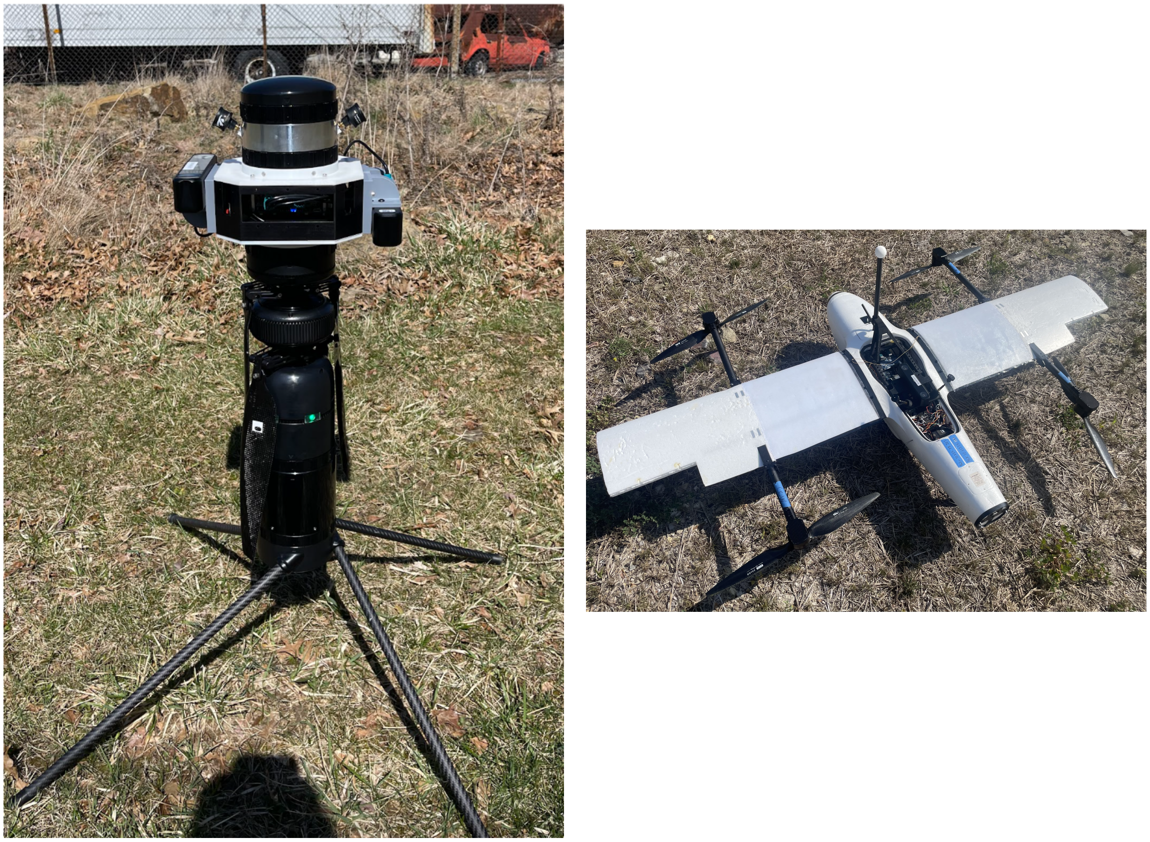


Figure 4.6: Left: A coaxial drone wire data collection platform, Right: a VTOL wire data collection platform

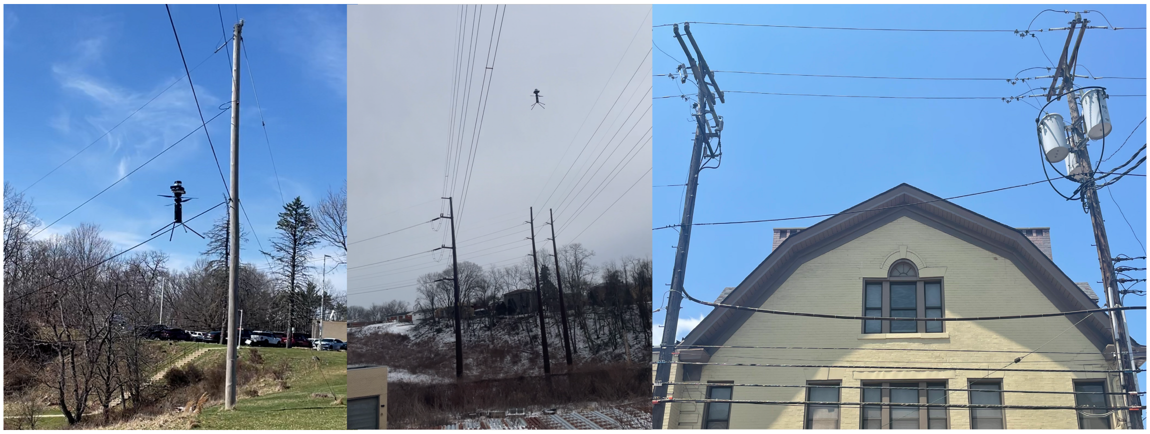


Figure 4.7: Real-world multi-wire data collection locations

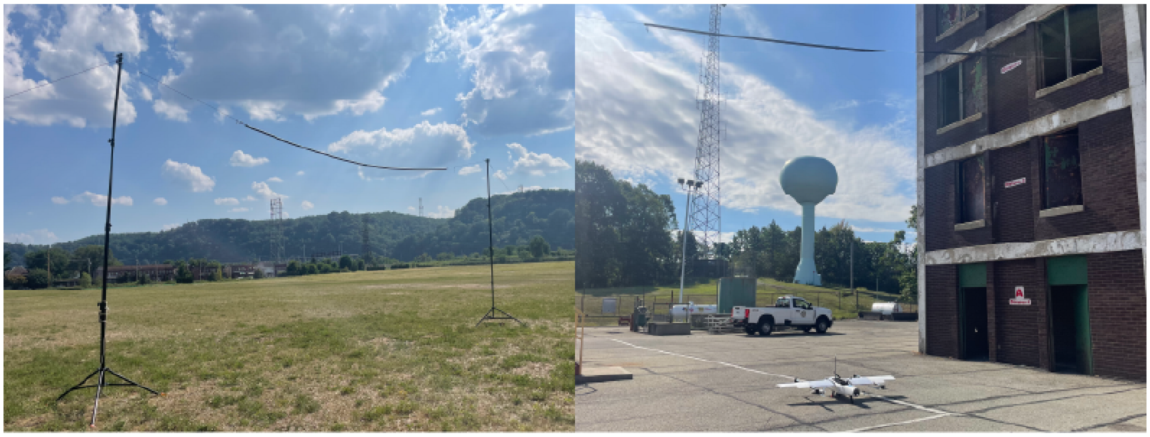


Figure 4.8: Real-world pseudo single-wire test setups

Chapter 5

Results

Across all testing environments including controlled indoor experiments, outdoor field trials, and high fidelity simulation, the goals of our evaluation were consistent: to demonstrate that a fully passive, camera only wire tracking system can achieve multi wire tracking and approach. The architecture inherently satisfies the first three requirements, including ubiquity, multi wire capability, and lightweight implementation, through its software and hardware design. However, the final two requirements, accuracy and real time operation, remain to be validated and are assessed along a spectrum of performance.

Using the three testing environments, indoor, outdoor, and simulation, we aim to demonstrate that the system achieves sufficient accuracy to enable real world wire approach with reliable position estimates, and that the algorithm can run on edge compute carried onboard the drone while maintaining a fast enough update rate to respond to environmental or pose estimation errors. These testing environments allow us to evaluate performance under varying levels of uncertainty, from the controlled conditions of simulation to the most challenging scenarios encountered outdoors.

5.1 Wire Detection Results

Overall, we aim to evaluate the accuracy of wire detection across various scenarios to determine the limits of our method. Using the indoor testing setup, we can test against ground truth and benchmark the algorithm to assess detection accuracy.

For our first experiment, we wanted to directly compare our passive sensing approach against alternative sensor fusion methods and quantify how a camera only pipeline performs relative to emissive sensing. In Table 5.1, we present a comparison between our stereo based wire detection system and a mmWave radar at a distance of one meter.

Figure 5.1: Upward view of mmWave radar versus stereo camera detection for a 1.5in wire at 1 meter

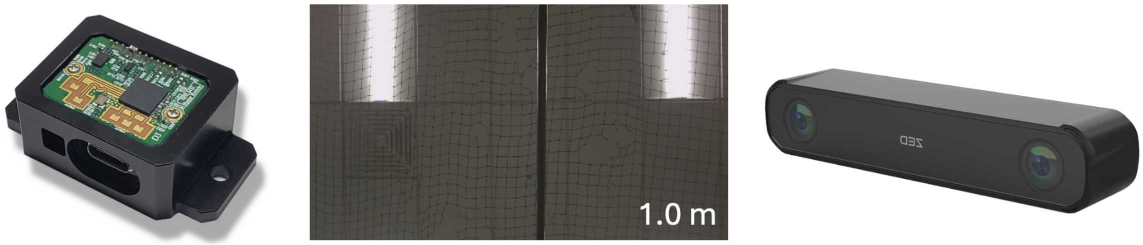


Table 5.1: Accuracy comparison between mmWave Radar and stereo camera wire detection on a 1.5in wire at 1 meter

Axis	**mmRadar Mean (m)**[a]	**mmRadar Variance (m)**[a]	**Stereo Mean (m)**	**Stereo Variance (m)**
X	-0.026	$2.425 \cdot 10^{-5}$	—[c]	0.025[b]
Y	-0.023	$1.098 \cdot 10^{-5}$	—[c]	0.025[b]
Z	0.982	$8.015 \cdot 10^{-6}$	1.029	0.026

[a] Values referenced from [9].

[b] With our representation, there is no x and y, only a horizontal distance from the wire angle.

[c] In our data collection, our setup was not conducive to centering the wire above

As expected, the mmWave radar demonstrates slightly lower mean error and notably smaller variance across all axes, reflecting the inherent precision of emissive depth sensing modalities. Despite this, the stereo RGB camera system achieves mean error values that are surprisingly comparable in practice, even though its variance is higher. This result highlights that a purely passive, non emissive method can still deliver accurate wire position estimates suitable for close range operation. These findings suggest that camera only perception can serve as a viable alternative to mmWave radar in applications where low mass, non-emissivity, and simplicity are prioritized. Additionally, it highlights the need for a method, that can take out the variance seen both in other papers like [8] and this experiment.

In a similar vain, we wanted to see how this depth estimation at one meter degraded as the camera moved farther from the wire. We conduct an experiment in Table 5.2 where we see the depth estimation values at 1.0, 2.2, and 5.27 meters.

Figure 5.2: Under wire view of a 1.5in wire at varying heights

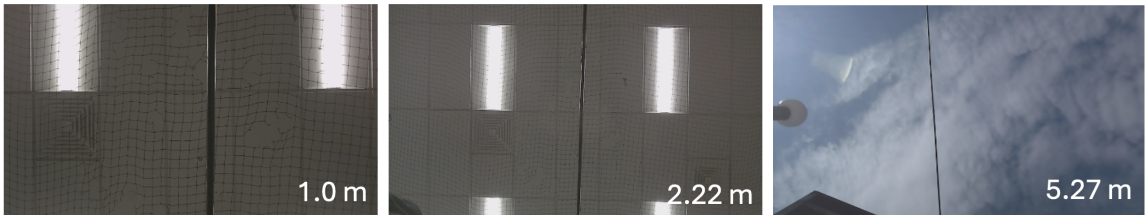


Table 5.2: Height accuracy across varying camera-to-wire distances

Z Height (m)	**Mean Z (m)**	**Mean Error Z (m)**	**Var Z (m)**
1.0	1.029	0.029	0.026
2.22	2.25	0.030	0.072
5.27	5.43	0.160	0.288

From Table 5.2, we observe a predictable trend: as the distance to the wire increases, both the mean error and the variance grow. At close range (1.0 m and 2.22 m), the mean vertical error remains very small, around three centimeters, with only a modest increase in variance. Even at the farthest tested distance of 5.27 m, the mean error rises to approximately sixteen centimeters, with a corresponding increase in variance. Despite this degradation, the overall accuracy remains well within the tolerance required for wire approach, as servoing relies on maintaining decimeter-level precision rather than exact millimeter alignment. These results confirm that the stereo-based system provides sufficiently stable and accurate distance estimates across the full range of expected operating heights, enabling reliable positional servoing to a wire.

We also evaluated how depth accuracy is affected as the wire becomes more parallel to the camera. Since stereo cameras rely on disparity between the left and right images, wires that are nearly parallel to the camera pose a greater challenge for correct pixel matching. Table 5.3 shows how the relative angle between the stereo camera and the wire impacts the accuracy of the wire detection algorithm, highlighting reduced performance at more extreme angles.

Figure 5.3: Under wire view of a 1.5in wire at varying angles

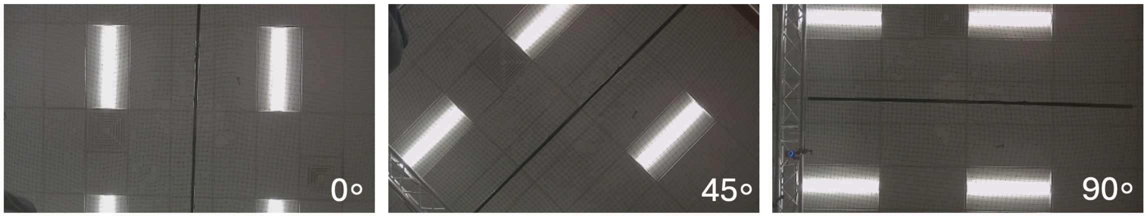


Table 5.3: Benchmark metrics for wire detection performance at different wire angles at 2.22 meters

Metric	**0°**	**45°**	**90°**
Mean Vertical Distance (m)	2.252	2.336	2.357
Ground Truth Vertical Error (m)	0.068	0.114	0.138
Std Dev Vertical Distance (m)	0.072	0.053	0.052
Std Dev Horizontal Error (m)	0.037	0.184	0.295
Yaw Deviation (deg)	0.259	0.333	2.598

Across all tested orientations, the system demonstrates robust vertical distance estimation, with mean error remaining low and only a slight increase as the wire angle approaches 90°. The vertical distance standard deviation remains consistently small, indicating strong frame-to-frame stability in depth and line fitting performance. In contrast, horizontal localization and directional accuracy exhibit clear angle-dependent degradation. As the wire rotates from 0° to 90°, the horizontal error increases by nearly an order of magnitude, and yaw deviation rises sharply, reflecting the geometric ambiguity introduced when wires appear nearly vertical in the camera frame.

These results indicate that, while horizontal position and yaw estimation degrade for near-vertical wires due to limited horizontal image structure, the resulting error remains within acceptable bounds for wire approach. Aligning the drone approx-

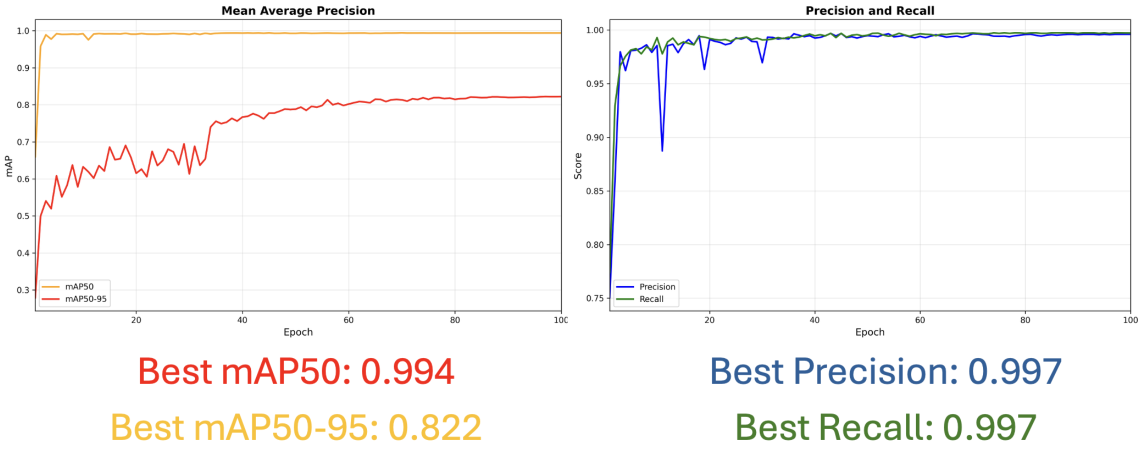


Figure 5.4: Wire detection model training metrics

imately with the wire before servoing further mitigates this limitation, ensuring reliable orientation and successful wire tracking. Overall, these findings demonstrate that our wire detection method is comparable to existing approaches and provides sufficient accuracy for practical wire approach tasks.

5.2 Detection Model Results

To improve region-of-interest selection, we also trained a model capable of detecting instance-level ROIs directly in each camera frame. This was made possible by automatically generating a large labeled dataset using our classical detection pipeline. In total, we aggregated 11,212 images with corresponding ground-truth bounding boxes to fine-tune a YOLO11 Oriented Bounding Box model. Figure 5.4 presents the key evaluation metrics used to assess the model's performance and overall detection accuracy.

Even with aggressive augmentations applied to increase data diversity, the training results are highly encouraging. The model demonstrates strong detection performance, achieving a precision and recall of 0.997, indicating its ability to reliably identify true positives while minimizing false negatives. With a mean average precision (mAP@50–95) of 0.822, the model also shows high confidence in its predictions. While these metrics suggest strong tuning to the training distribution, it remains important to validate performance across real-world frames.

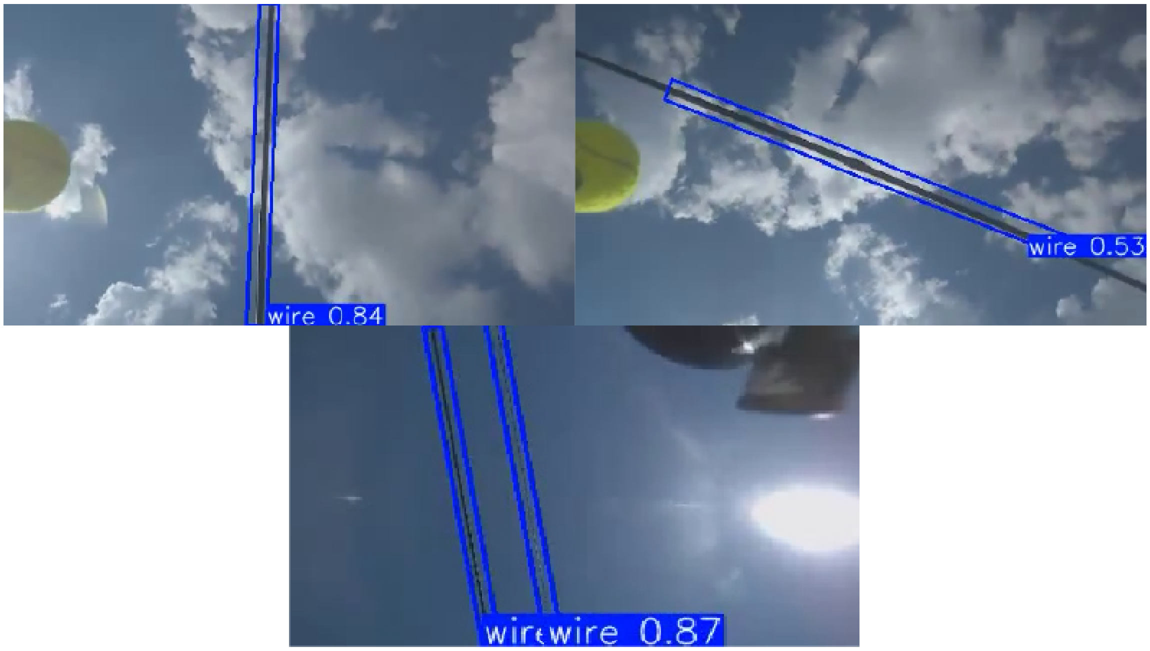


Figure 5.5: Inference examples generated by the fine-tuned YOLO11 oriented bounding box model

Figure 5.5 illustrates model inferences on a varied set of images, including single straight wires, rotated wires, and multi-wire scenes. Although the training set contained only a limited number of multi-wire examples due to the difficulty of collecting clean data for such scenarios, the model still generalizes well to these out-of-distribution cases, as demonstrated in the multi-wire example shown in Figure 5.5.

Additionally, when comparing the performance of the model-based detector to our classical pipeline, Table 5.4 highlights the model's substantial accuracy gains, particularly in challenging scenarios where the classical approach struggles.

Table 5.4: Inference accuracy of our classical detection versus our model-based detection in different wire scenarios

Method	**Accuracy per Detection (%)**	
	Single Wire	**Double Wire**
Classical	92.71	100.00
Model	0.00	36.36

In the single-wire scenario, the classical pipeline remains reasonably accurate but can still occasionally miscount, detecting zero or two wires due to visual ambiguity or depth noise. In contrast, the model achieves nearly perfect performance, consistently identifying the single wire with close to 100% accuracy.

In the multi-wire scenario, the limitations of the classical approach become more pronounced: intense sun glare, reduced image resolution and contrast, and highly degraded depth measurements prevent it from reliably detecting either wire. The model, however, maintains substantially higher accuracy, correctly identifying the number of wires in the scene far more frequently.

These results demonstrate that incorporating the model significantly improves the reliability of the wire detection stage of the pipeline. This suggests that model-based detection can meaningfully augment, or eventually replace, components of the classical pipeline in future research.

5.3 Wire Tracking Results

The primary validation points for our system are accuracy and real-time performance, particularly with the wire detection algorithm augmented by the wire tracking layer. To evaluate runtime performance, we conducted a benchmark test using a handheld jig with a single wire in the indoor testing setup. The full perception pipeline was profiled on the NVIDIA Jetson Orin AGX 64GB to quantify real-time performance on an embedded edge compute platform.

Table 5.5: Performance benchmarks for wire detection and tracking

Metric	**Results**
Inference Speed (FPS)	42.84
Inference Time (ms)	24.34
CPU Usage (%)	37.96
GPU Usage (%)	68.79

As shown in Table 5.5, the algorithm achieves nearly 43 FPS, with an average inference time of just 24 ms. This demonstrates that detection and tracking operate consistently at high frequency, exceeding the native 30 FPS of most onboard cameras. High-rate position estimates from the UAV state estimator can further complement this performance, mitigating any latency introduced by the inference cycle. Moderate

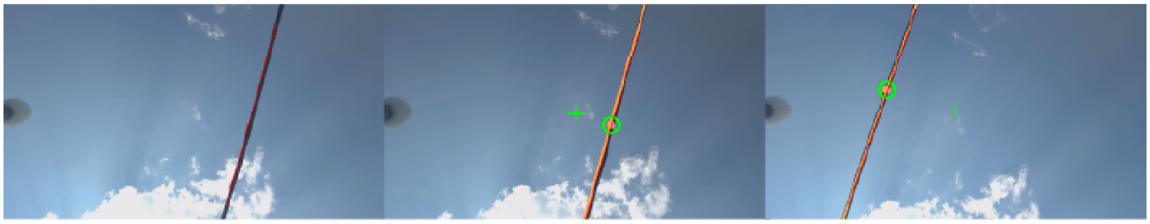


Figure 5.6: Frames from a video sequence demonstrating our wire tracking method on a single-wire scenario. The leftmost image shows the wire beginning to be detected but not yet tracked, while the final two images show the track initialized and maintained across frames

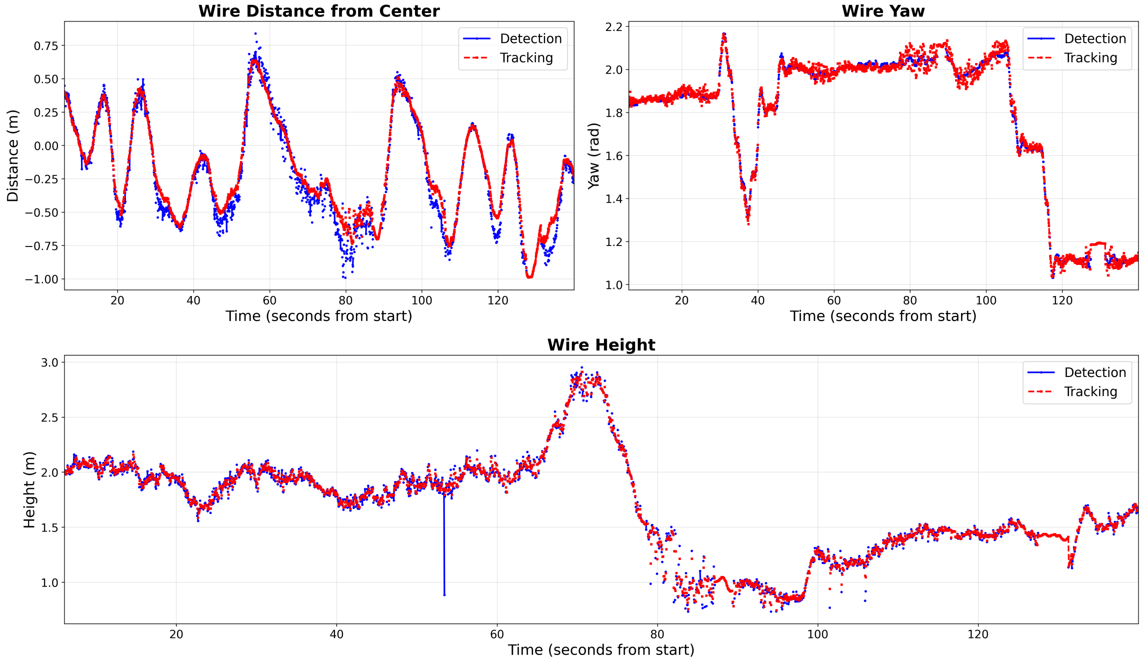
CPU and GPU utilization indicates that the system consumes only a portion of the available compute resources, leaving headroom for additional autonomy tasks such as control, planning, and communication. These results confirm that our approach is both computationally efficient and capable of supporting real-time wire tracking and servoing in real-world deployment scenarios.

To evaluate real-world efficacy, we tested the wire tracking algorithm on one of our outdoor data runs. Figure 5.6 shows multiple frames where a wire is first detected, then successfully initialized and tracked across frames. The green circle indicates the actively tracked wire instance in the camera frame.

We can qualitatively assess how wire tracking improves upon wire detection, and we can also visualize the smoothing effect that tracking provides by filtering noisy and inaccurate measurements. In Figure 5.7, wire tracks and wire detections are plotted along the distance, yaw, and height dimensions.

In all three graphs, the smoothing behavior of the tracking method is evident. The tracked points, shown in red, are smooth and visually continuous, while the raw detections are scattered and noisy. This demonstrates that the tracking algorithm effectively suppresses the high variance inherent in stereo depth measurements. Additionally, a major outlier in the height dimension is neutralized by the tracking algorithm, which identifies it as inconsistent with the predicted wire position.

Overall, these results show that the wire tracking algorithm not only maintains consistent wire instance associations across frames but also improves detection accuracy by filtering noise and removing outliers beyond the expected motion range of the wires.


Wire Distance from Center

Wire Yaw

Wire Height

Figure 5.7: Comparison of noise between raw wire detections and tracked wire instances.

5.4 Wire Approach Results

To evaluate the effectiveness of wire tracking for approaching a selected instance, we tested the system in a safe, controlled simulation environment before attempting real-world deployment. Simulation allowed us to assess the visual and autonomy aspects of the system, as perception and control closely mimicked real-world scenarios while providing more accurate position estimates and control through tunable position noise than what was achievable outdoors. The simulator was designed with the same interfaces as a real drone, so aside from tuning constants, the same visual servoing algorithm could be applied in both indoor and outdoor environments.

The simulated PID-based servoing framework directly mirrors the control logic implemented in hardware. In the physical system, the node receives wire positions from the detection and tracking pipeline, applies Cartesian offsets, and continuously computes velocity commands using PID loops for x, y, z, and yaw, which are sent to the UAV via MAVROS. The simulation thus serves as a testbed for evaluating

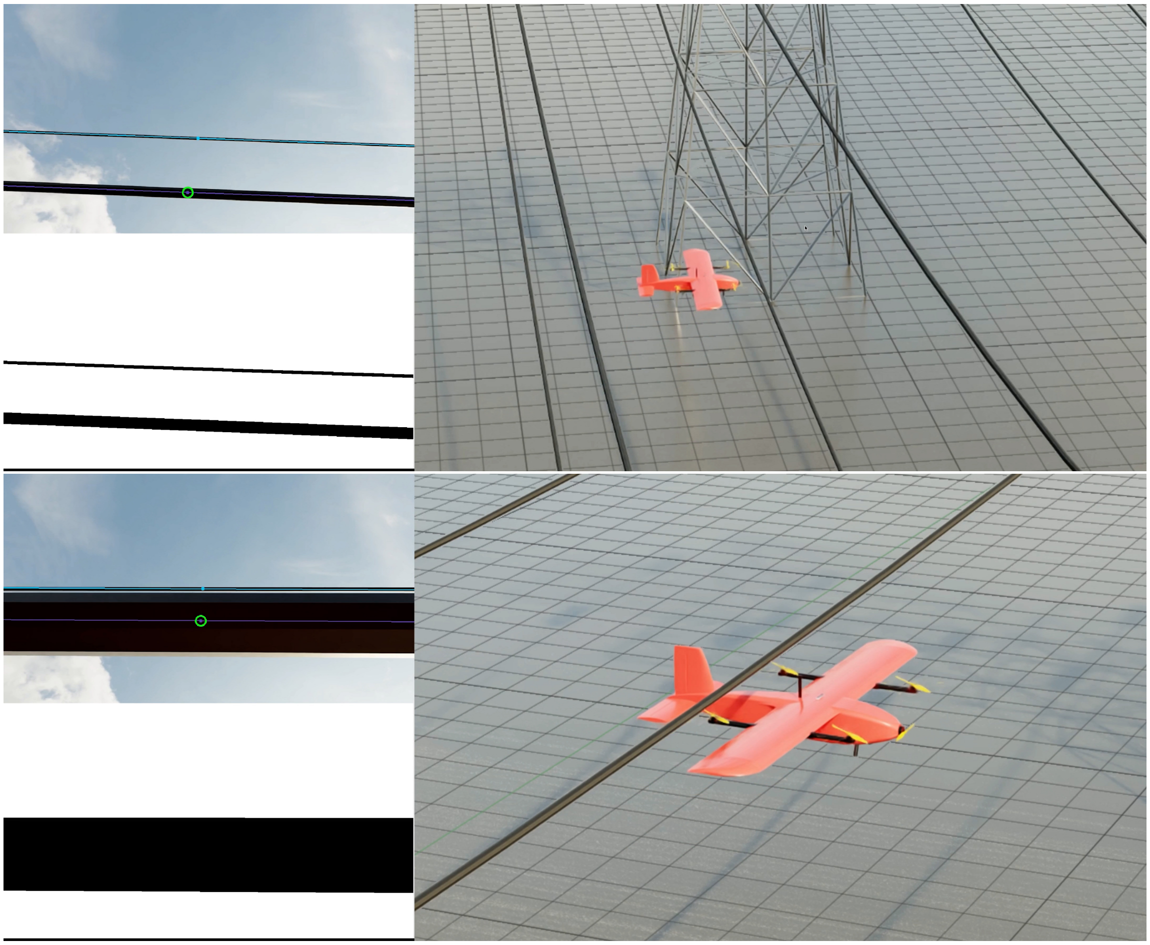


Figure 5.8: Frames from a video of the VTOL model servoing to a selected wire (circled in green) in simulation

control stability and validating approach strategies, ensuring a smooth transition to real-world autonomous wire approach.

As shown in Figure 5.8, the VTOL detects multiple wire instances and initializes tracks for each. The consistent coloring between the top and bottom frames confirms that instances are correctly attributed in each frame. The green circle indicates the wire instance that the VTOL is actively tracking. In the bottom frame, the green circle is centered with a small offset, demonstrating the PID PBVS in action: the system takes estimated wire positions and computes real-time control commands to achieve the desired approach position.

This demonstration highlights the end-to-end capability of the system and validates the framework for future real-world tests. One significant limitation for outdoor deployment is the drone's positional accuracy. Preliminary outdoor tests showed that insufficiently precise state estimates caused drift, making reliable wire approach challenging. Future work will focus on improving the hardware platform to provide more accurate state estimation.

5.5 Overall System Comparison

Finally, Table 5.6 compares our approach with existing powerline and wire tracking systems, highlighting key distinctions. Most prior work relies on active sensing hardware, such as multi-segment LiDAR arrays or radar, or can track only a single large object in vision-based systems. In contrast, our method is the only purely passive, camera-only system capable of reliable multi-wire tracking, including wires smaller than 4 cm. Despite using simpler sensing hardware, our pipeline achieves performance comparable to systems with more complex or fused sensor suites.

Table 5.6: Comparison of related work on wire tracking

Category	**LineDrone** [11]	**LOCATOR** [5]	**Thomas et al.** [14]	**Ramon-Soria et al.** [12]	**Hoang et al.**[4]	**Ours**
Sensors	RGB Camera + 2D LiDAR	2 × 8-segment 2D LiDARs	RGB Camera	RGB + stereo camera	RGB camera + mmWave radar	Stereo RGB camera
Type of tracked object	Powerline	Powerline	Pipe	Pipe	Powerline	Powerline
Object diameter	1.9–3.8 cm	3.8 cm	15 cm^a	15 cm^a	1 and 2 cm	2.54-3.8 cm
Number of tracked objects	1	1	1	1	Multiple	Multiple
Processing system	CPU+GPU	CPU	CPU	CPU+GPU	CPU+FPGA	CPU+GPU
Approach object from	Above	Below	Below	Above	Below	Below
Perception update rate	30 Hz	–	75 Hz	16 Hz	100 Hz	43 Hz
Perception range	2.5 m	5 m	–	–	10 m	6 m

Chapter 6

Conclusions

6.1 Summary

In summary, this work introduces the first passive-sensing approach capable of simultaneous multi-wire detection and tracking, addressing a longstanding challenge in safe and precise aerial powerline approach. The method achieves real-time performance with high accuracy, comparable to systems relying on active depth sensing or situational assumptions. Extensive validation in indoor experiments, outdoor field tests, and high-fidelity simulation confirms the system's robustness and generalizability. These contributions establish a reliable foundation for vision-based wire tracking in complex environments, demonstrating a multi-wire instance detection and tracking system suitable for deployment on lightweight aerial robots.

6.2 Limitations

Several important limitations remain before the proposed system can support fully autonomous wire approach and charging. First, real-world servoing to a wire has not yet been demonstrated, largely due to insufficiently accurate position estimates on the current platforms. This restricts the ability to validate closed-loop control in practical settings. Second, long-range wire approach beyond ten meters remains unproven. At such distances, wire position estimates are expected to degrade due to reduced visual resolution, increased depth noise, and the inherently thin and low-texture nature of wires, all of which may impact detection and tracking performance. Finally, a complete wire-attachment mechanism has not yet been developed, and the attachment and charging process must be integrated with the wire-approach behavior to form a unified system. Addressing these limitations will be critical for enabling end-to-end autonomous wire engagement in the real-world.

6.3 Future Work

Future work will focus first on improving the reliability of real-world servoing through enhanced localization. The current platform lacks the positional accuracy required for stable closed-loop control during wire approach. Incorporating higher-fidelity localization methods, such as RTK-GNSS or improved visual odometry pipelines, would provide the precise state estimates necessary to support position-based control.

A second direction involves extending the effective range of wire detection and approach beyond ten meters. At long distances, passive stereo depth estimation becomes unreliable due to the small image footprint and low texture of wires. Future work could incorporate long-distance wire detection models to overcome these limitations. Methods based on ray-tracing inference or monocular depth cues, such as those explored in [1], may help improve robustness when wires appear small or distant in the field of view. In addition, fusing passive vision with complementary sensing modalities, such as Hall-effect sensors, could enhance performance under challenging lighting conditions, rapid motion, or environments where stereo depth cues degrade. Lastly, to extend range, developing a hybrid control strategy that transitions from image-based visual servoing (IBVS) at longer range to position-based visual servoing (PBVS) when wire localization estimates are sufficiently accurate would improve system stability across varying operational distances.

Finally, to enable end-to-end autonomous wire engagement, further development of hardware and attachment mechanisms is required. Designing a reliable gripper or attachment device is essential for physical interaction with the wire. Beyond hardware, a coordinated attachment and charging sequence must be developed and integrated with the perception and servoing pipeline to support fully autonomous operation. Collectively, these efforts would expand the capabilities of the system and move it closer to long-range autonomous wire approach, attachment, and power transfer.

Bibliography

[1] Omar Alama, Avigyan Bhattacharya, Haoyang He, Seungchan Kim, Yuheng Qiu, Wenshan Wang, Cherie Ho, Nikhil Keetha, and Sebastian Scherer. Rayfronts: Open-set semantic ray frontiers for online scene understanding and exploration, 2025. URL https://arxiv.org/abs/2504.06994. 6.3

[2] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 1986. doi: 10.1109/TPAMI.1986.4767851. 3.2.1

[3] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981. ISSN 0001-0782. doi: 10.1145/358669.358692. URL https://doi.org/10.1145/358669.358692. 3.2.3

[4] Viet Duong Hoang, Frederik Falk Nyboe, Nicolaj Haarhøj Malle, and Emad Ebeid. Autonomous overhead powerline recharging for uninterrupted drone operations. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13551–13557, 2024. doi: 10.1109/ICRA57147.2024.10611522. 2.1.3, 5.6

[5] Nicolai Iversen, Oscar Bowen Schofield, and Emad Ebeid. Locator - lightweight and low-cost autonomous drone system for overhead cable detection and soft grasping. In *2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 205–212, 2020. doi: 10.1109/SSRR50563.2020.9292591. 2.1.1, 5.6

[6] Rahima Khanam and Muhammad Hussain. Yolov11: An overview of the key architectural enhancements, 2024. URL https://arxiv.org/abs/2410.17725. 3.2.4

[7] Angus Lynch, Corey Duguid, Joao Buzzatto, and Minas Liarokapis. A powerline inspection uav equipped with dexterous, lockable gripping mechanisms for autonomous perching and contact rolling. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6206–6211, 2024. doi: 10.1109/ICRA57147.2024.10610783. 2.1.2, 3.2.2

[8] Nicolaj Haarhøj Malle, Frederik Falk Nyboe, and Emad Ebeid. Survey and evaluation of sensors for overhead cable detection using uavs. In *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 361–370, 2021. doi: 10.1109/ICUAS51884.2021.9476724. (document), 2.2, 2.4, 5.1

[9] Nicolaj Haarhøj Malle, Frederik Falk Nyboe, and Emad Samuel Malki Ebeid. Onboard powerline perception system for uavs using mmwave radar and fpga-accelerated vision. *IEEE Access*, 10:113543–113559, 2022. doi: 10.1109/ACCESS.2022.3217537. 2.1.3, 3.5, **??**

[10] J. Matas, C. Galambos, and J. Kittler. Robust detection of lines using the progressive probabilistic hough transform. *Computer Vision and Image Understanding*, 78(1):119–137, 2000. ISSN 1077-3142. doi: https://doi.org/10.1006/cviu.1999.0831. URL `https://www.sciencedirect.com/science/article/pii/S1077314299908317`. 3.2.1

[11] François Mirallès, Philippe Hamelin, Ghislain Lambert, Samuel Lavoie, Nicolas Pouliot, Matthieu Montfrond, and Serge Montambault. Linedrone technology: Landing an unmanned aerial vehicle on a power line. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6545–6552, 2018. doi: 10.1109/ICRA.2018.8461250. 2.1.3, 5.6

[12] P. Ramon-Soria, A.E. Gomez-Tamm, F.J. Garcia-Rubiales, B.C. Arrue, and A. Ollero. Autonomous landing on pipes using soft gripper for inspection and maintenance in outdoor environments. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5832–5839, 2019. doi: 10.1109/IROS40897.2019.8967850. 2.1.2, 5.6

[13] RTI. Real-time innovations (rti) dds standard. `https://www.rti.com/products/dds-standard`. [Online; accessed 19-July-2025]. 3.2

[14] Justin Thomas, Giuseppe Loianno, Kostas Daniilidis, and Vijay Kumar. Visual servoing of quadrotors for perching by hanging from cylindrical objects. *IEEE Robotics and Automation Letters*, 1(1):57–64, 2016. doi: 10.1109/LRA.2015.2506001. 2.1.2, 5.6

[15] Miaowei Wang and Daniel Morris. Self-annotated 3d geometric learning for smeared points removal. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3494–3503, January 2024. 2.4