SPAICE Technical Challenge: 3D Tracking of a Single Drone

Nicholas Morrison
October 2025

0. Introduction

This report proposes a multi-camera vision system to detect and track a single flying drone in real-time. The pipeline uses YOLO for 2D detection, triangulation for depth estimation, and a Kalman filter for temporal tracking. The goal is to provide accurate 3D localization of the drone across complex trajectories, while remaining efficient and robust to challenges like occlusion and background clutter. Further extensions including thermal/RGB fusion and non-visual sensor integration are also discussed. A proof-of-concept implementation is demonstrated using the ETH multi-view drone dataset, with code available on Github.

1. System Design Overview

The proposed system uses a network of fixed, monocular RGB cameras positioned around the monitored area. A wide-angle camera, such as a GoPro, is placed centrally to maximize coverage and ensure that the drone remains within the field of view. Additional cameras with standard lenses are distributed at different locations to provide high-quality video feeds with overlapping perspectives, which are necessary for triangulation of the drone's 3D positions. The cameras remain static rather than mounted on gimbals, as fixed viewpoints ensure consistent geometry. Although gimbaled systems could help maintain the target in view, they are less suitable for tracking fast, erratic motions and can introduce additional complexity to calibration.

Depth sensing is considered across several modalities. Stereo rigs, a pair of side-by-side lenses like human eyes, can estimate depth through parallax. However, accuracy deteriorates at long range due to small disparities. RGB-D sensors, which use active depth sensing through time of flight or structured light, are effective indoors, but have limited range which leads them to fail outdoors. Hyperspectral or multispectral sensors are powerful but too costly and slow for real-time deployment. For drone monitoring, where range and robustness are critical, a multi-camera RGB network remains the most practical solution, supplemented by strong temporal tracking to bridge gaps when detections are lost.

Calibration is crucial to the system's accuracy. Each camera's intrinsic parameters are obtained using standard checkerboard calibration, which allows estimation of focal length, principal point, and distortion coefficients. This step is particularly important when using wide-angle lenses like GoPro's, where fisheye distortion must be corrected. Extrinsic calibration, which defines the relative poses between cameras, is established from known positions in a common coordinate frame which can be derived from total-station measurements. Finally, since cameras can operate at different frame rates and offsets, synchronization is necessary to align frames for triangulation so that they represent the same instant in time.

2. Algorithmic Approach

The pipeline consists of three main stages: detection, 3D position estimation, and tracking. For this process, I relied on a combination of traditional foundation models from university classes and further research to determine what SOTA methods people are currently using in industry.

2.1 Detection

The system uses YOLO (You Only Look Once), a lightweight CNN detector. YOLO is well-suited for this application as it achieves real-time inference speeds while maintaining performance on small objects such as drones by learning general representations. The neural network predicts bounding boxes and class probabilities directly from the whole image in one single-shot evaluation.Traditional alternatives, such as background subtraction or optical flow struggle with dynamic outdoor environments, with moving clouds and lighting changes. Other simple approaches like patch similarity or correlation filters are easy to implement, but lack robustness to rotation and occlusion. On the other end of the spectrum, more advanced transformer-based architectures, such as DETR, can be more accurate with unseen drone types but are computationally heavy for edge deployment. YOLO therefore provides the best balance between speed and robustness.

2.2 Depth Estimation

Once the drone is detected in 2D in at least two camera views, its 3D position can be estimated using epipolar geometry. Given known intrinsics, the essential matrix is estimated from corresponding detections using RANSAC and the eight-point algorithm. The essential matrix encodes the relative rotation and translation between two cameras, which can be decomposed to recover their relative pose. With the camera intrinsics and extrinsics known, each 2D detection defines a projection ray in space, originating from the camera center and extending through the pixel coordinates. Triangulation estimates the drone's 3D position by solving for the point of closest approach between these rays. Since noise prevents exact intersection, the solution is obtained using least squares or singular value decomposition. When multiple cameras observe the drone simultaneously, the reconstruction becomes more robust. Methods such as bundle adjustment can refine accuracy by optimizing across all cameras jointly.

2.3 Tracking

The estimated 3D points are then passed through a Kalman filter for temporal tracking. The Kalman filter provides a lightweight and efficient means of maintaining a stable trajectory, even when detections are noisy or occasionally missing. It predicts the drone's position and velocity between detections and corrects its estimate when new measurements are available. In practice, I use the SORT algorithm which combines the Kalman filter and Hungarian algorithm. For the single-drone case, a standard linear Kalman filter is sufficient, although extended or unscented variants could be applied for more complex non-linear motion models. One downside is assumption that the posterior be represented as a normal distribution, which is where particle filters like condensation come in handy to represent multi-model distributions. Alternative deep learning-based methods like DeepSORT, were considered but add complexity unnecessary for a single-object scenario with live tracking.

3. Failure Modes & Robustness

Real-world conditions pose several challenges that must be addressed for reliable operation. Occlusion is inevitable when the drone passes behind obstacles or temporarily leaves a camera's field of view. If the drone is only in one camera in the system FOV, we must rely on the tracking algorithm. The Kalman filter mitigates this by maintaining an estimate until detections resume. Motion blur reduces detection accuracy which can be alleviated through appropriate camera settings, such as high shutter speeds.

While my pipeline works well in daytime conditions, For low-light and nighttime situations, the system can be augmented with thermal or infrared cameras. At long ranges, the drone may occupy only a handful of pixels, which makes detection difficult. A hybrid setup combining wide-angle and zoom cameras would help: wide-angle cameras provide broad coverage, while gimbaled zoomed-in cameras offer detailed observations once the drone has been located.

In terms of latency, these pipelines need to work in real time on edge devices. My proof of concept works by analyzing the frames of a prerecorded video. In the real world, it would likely be necessary to reduce the resolution or only do inference on N frames, or perhaps pick a lighter YOLO version. Also we need to ensure synchronization or the triangulation will fail. Also if there are multiple drones or other flying objects, the model should be able to handle it if YOLO is finetuned on the drone dataset. Though perhaps DEEPsort or Strong-sort would be better, even if heavier, to handle target crossings and cluttered backgrounds.

4. Further Improvements

The system described above relies solely on visual information, but additional sensing modalities could greatly improve robustness. As mentioned, thermal infrared cameras can complement RGB cameras by providing contrast between drones and the background in low-light conditions. Acoustic sensors could detect the characteristic sound of drone rotors, allowing localization even when the drone is not visible. Radar offers long-range detection unaffected by weather or lighting and could provide an initial cue for the visual system to focus on. Sensor fusion can be implemented using a Kalman filter with multiple measurement models or through probabilistic Bayesian approaches, combining cues from different modalities to produce more reliable state estimates.

5. Proof of Concept

For the prototype implementation, I use the ETH multi-view drone dataset which includes video footage from a ground visual system as well as ground truth 3D trajectories of drones recorded by an RTK system. https://github.com/CenekAlbl/drone-tracking-datasets?tab=readme-ov-file

The code can be found at this github repo: https://github.com/nmorrison01/drone-tracker