Dynamic Multiple Object Tracking from a UAV

EGH400-2

Darren Gebler
Dr. Simon Denman

Submitted: 07 of November 2021

Abstract

This thesis presents an efficacious method to track animals detected from thermal videos using an object detection model developed by Corcoran et al. The technique implemented is suitable for tracking dynamic animals from a moving camera. Existing tracking methods implemented employs the nearest neighbour method, which assigns previous detections to the closest new observation. This poses issues when tracking multiple animals. The process to track multiple dynamic objects first converts pixel coordinates into a northeast down coordinate system. Then, as all detections are on the same coordinate plane as the UAV, an objects state can begin to be estimated using a Kalman filter. This ensures that it can be reassigned when a detection goes out of the frame if it is observed again. A global nearest neighbour approach to data association was implemented to ensure clusters of animals are correctly assigned when observed across frames. Data association is vital as state estimation relies upon subsequent detections to adapt to changing velocity and heading.

Acknowledgements

First and foremost, I would like to extend my gratitude and appreciation to my supervisor, Dr Simon Denman. Simon offered me many opportunities to lay my path and provided endless support, inspiration and guidance. I appreciate your patience and prompt feedback, and I have gained an incredible amount of knowledge and respect for this research field. It was an honour completing my final research project with your support, and I am glad you provided a smooth and seamless experience through this challenging and untried time.

Contents

List of Figures

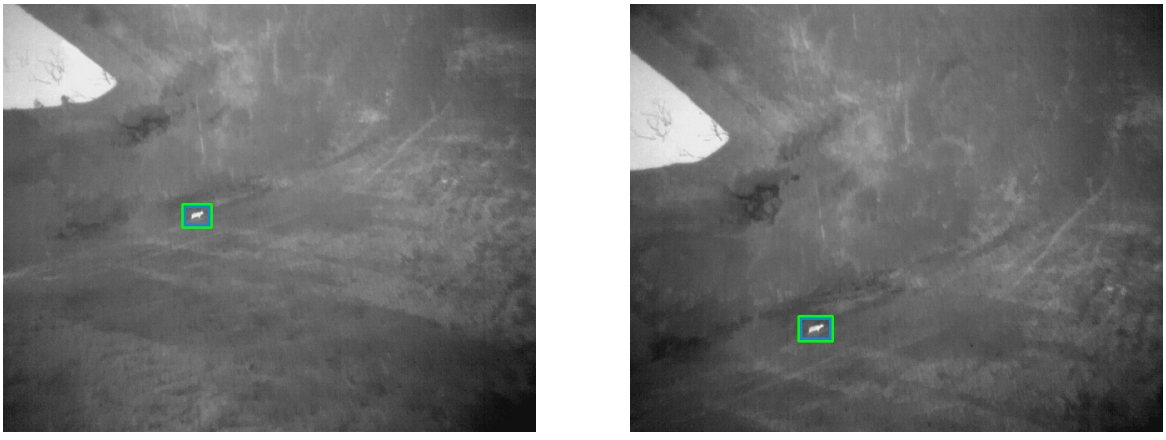
List of Tables

1 Introduction

Wildlife abundance monitoring is essential in ensuring threatened animals presence and distribution are tracked and do not slide towards extinction without notice [16]. However, traditional monitoring methods such as 'boots on the ground' are labour-intensive and typically slow and inaccurate depending on conditions [13]. With the recent machine learning (ML) boom, it is natural that efforts have been made to automate wildlife monitoring.

Corcoran et al. developed an automation method for detecting and monitoring Koalas using low-level aerial surveillance and machine learning [8]. Low-level aerial surveillance is achieved using a DJI Matrice 210 (UAV), attached with a FLIR thermal camera to capture imagery to be processed. With the success of Evangeline's work, researchers, such as Dr Denman, realised that this method could be adapted to detect any animal. So long as an appropriate ML model is trained and the animal is warm-blooded (detection is achieved using heat signatures). The most notable new detection model detects deer. However, this new detection model poses an issue with abundance counting and monitoring. Previously, it was safe to assume that the koalas detected will not move as a drone flies over an area of interest. Unfortunately, this same assumption can not be made for deer as they move unpredictably. Understanding why this causes issues in the original counting method is essential to understand how data is captured.

After a section of land is identified by a wildlife conservationist, a drone pilot will fly in a lawnmower pattern (example, south to north, then north to south) at a fixed altitude above the area of interest. As the drone flies, thermal data is captured at eight frames per second (FPS) from a stabilised gimbal. Once an animal comes into frame, it is more than likely to stay in frame for some time, so the same observation must not be counted more than once. An example is shown in Figure 1 below.



(a) First Detection Frame *(b) Second Detection Frame*

Figure 1: Example of the same object being detected across frames. The green box represents a detection.

Currently, this problem is solved using the nearest neighbour method, where the closest subsequent detection is assigned to the latest observation, given some threshold. However, the observation is detected 500 frames again from the last detection after being out of the camera's field of view (FOV), as shown in Figure 2.

Figure 2: Exact Detection 500 Frames after Last Detection

The current process cannot reidentify this object as the same one observed 67 frames ago and will therefore recount it. Understanding where an object is relative to the drone is vital for clusters of animals. For example, figure 3 below displays two subsequent frames with two different sized groups detected.

(a) Cluster One

(b) Cluster Two

Figure 3: Subsequent frames with a growing cluster of animals were detected

.

Associating the exact previous animal detection with its applicable new detection is vital to associate animals over time to improve counts accurately. In addition, correct association improves reidentification accuracy.

2 Literature Review

This section explores existing research in the fields of object tracking. Although the primary objective is to track detections accurately, further research was required as most previous literature uses a static camera model, where a static camera tracks dynamic objects. As the camera in this project is mounted to a dynamic UAV and detects dynamic objects, improvements were made to the previous research explored.

2.1 Georeference Transformation

As mentioned in Section 2 above, further research was required to accommodate the dynamic camera model used. When a detection is made, the returned coordinates of the object are in the camera coordinate frame. For example, with a thermal frame size of 640x512, an object detected in the middle of the frame will return the coordinates $x = 320$, $y = 256$. As the camera moves north, relative to the frame, the object y location will begin to shift south (assuming it does not move). This object can be tracked simply with a nearest neighbour assignment method. However, if this object goes out of the frame and returns into the frame at some time t, it is difficult to know if it is the same object detected previously. This is because the coordinates of the object were relative to the camera frame.

To solve the issue of object reidentification for static objects, detected coordinates are converted to a world coordinate frame. Gabrlik's [14] transformation of UAV attitude and position research describes a method called direct georeferencing. Direct georeferencing uses known exterior orientation measurements provided by the UAV's inertial measurement unit (IMU) and object detector to compute the absolute object point position. The exterior orientation measurements include 3 position coordinates $[X_0, Y_0, Z_0]$ and 3 attitude coordinates $[\phi, \theta, h]$. The cameras parameters, called interior orientation, are also required and are typically static variables defined in the camera specification. The most important is focal length, f. Absolute object point position is computed using a series of rotation and collinearity functions that transforms the camera system into a local system. Gabrlik's [14] research, however, requires every object to be visible in at least two overlapping frames. This is so the system can calculate the distance from the object to the UAV. This is not required for our research due to the reasons described in the following sections.

Wang et al. outlined a rapid georeference algorithm developed for emergency response [31]. Like Gabrlik's [14] research described above, it uses exterior and interior orientation measurements to transform the observed image into a world coordinate frame. Although the conversion follows a similar process, Wang et al. [31] defined an additional transformation step that converts the X_0, Y_0 and Z_0 geodetic coordinates to an earth-centred, earth-fixed (ECEF) frame. ECEF positions its origin at the centre of the earth, where X passes through the equator at the prime median, Z passes through the north pole and Y is in the equatorial plane perpendicular to X.

2.2 Object Tracking

Object tracking has become an important area of research in the computer vision field. Despite extensive research, issues arise from partial occlusion, object deformation, viewpoint changes and varying illumination. Many survey publications identify influential contributions that have shaped implementations of object trackers in object detecting algorithms. Notably, research by Zhan et al. [34], Hu et al. [17], Su Kim et al. [20], Candamo et al. [6], Wang [32] and Zhao et al. [36] describe key concepts that address the challenges as mentioned above.

2.2.1 Single Object Tracking

An object tracker is typically initialised when a new observation comes into the frame. A single object tracker must maintain the position of this observation across frames. A simple solution to track objects, single objects, and even multiple uncluttered objects, which Evangeline employs, is to use the nearest neighbour (NN) algorithm that assigns previous detections to the nearest subsequent observation. This solution successfully counts and tracks koalas, as typically, they are separated and do not move. Issues arise, however, as objects become cluttered.

2.2.2 Multiple Object Tracking

As this project focuses on improving abundance counting and tracking in deer and future animal detectors, multiple object tracking (MOT) is essential. One of the most challenging components of MOT is estimating trajectories of objects, including their re-entry and departure from the camera frame [1] [25]. Typical approaches to solve MOT combine a data association framework with an optimal state estimator to evaluate trajectories [2]. Data association attempts to measure the probability that a previous detection is the same in a subsequent frame. Optimal state estimators seek to predict an observations state in future frames. A Global Nearest Neighbour (GNN) approach encompasses both data association and state estimation algorithms to find and propagate the single most likely hypothesis at each scanned frame.

Pavlina Konstantinova et al. studied a GNN approach to multiple target tracking [22]. Observation track updating uses a procedure known as data correlation, which comprises two steps called gating and association [7] [3]. Gating is a test that eliminates unlikely observation-to-track pairings. First, a gate is formed around a predicted position. Then, object positions are predicted using a Kalman filter [19], which estimates the state of a linear system by learning over a series of observations that contain Gaussian noise and a known motion model. Association is required in dense target environments where an observation falls within the gates of multiple target tracks or when multiple observations fall within the gate of a target track. Optimal assignment minimises the summed distance for all individual assignments. This is achieved using Munkres assignment algorithm [27].

2.2.3 Machine Learning for Wildlife Monitoring

Many unique wildlife monitoring scenarios require optimised solutions to suit specific wildlife monitoring needs. Wildlife conservatives are looking for more cost-effective and efficient methods to monitor wild animals, so they are turning to machine learning. Due to this increased interest in the field, papers emerge with their unique solution to machine learning in wildlife monitoring.

Alexander Seymour et al. published a report introducing a solution to Automated detection and enumeration of marine wildlife using unmanned aircraft systems and thermal imagery [29]. It recognises that estimating animal populations is critical for wildlife management, specifically the abundance of two grey seal (Halichoerus grypus) breeding colonies in eastern Canada. Similar to E. Cocorcan's automated detection setup, A. Seymour utilises thermal imagery onboard a UAV to detect heat signatures in a specified area of interest. UAVs in wildlife monitoring is now widespread in modern wildlife monitoring techniques as they decrease cost and increase knowledge while reducing animal disturbance [28] [35].

As satellite image resolution exponentially improves, it has become feasible to assess wildlife populations from space [26]. However, issues arise as clouds and humidity can reduce accuracy and the difficulty to detect smaller animals.

2.2.4 Registration of Multi-Modal Data

Multimodal research is an emerging field in machine learning research. When developing a system, several different sensors are used to capture different information. Separately, this data can be used to make relatively simple predictions with a well-developed model. However, utilising all sensors and fusing the information will enable the development of more complex models to solve far more complex problems.

Frederik S. Leira et al. proposed a method for detecting, recognising and tracking boats from a UAV using a thermal camera [23]. The system uses a thermal camera to detect boats on a bed of water and assigns a track. Object tracking was done by using Kalman filters to estimate and predict the position and velocity. Further detections of the same object are used to update measurements passed to the Kalman filter. Without information fusion, however, object tracking would not be possible. Information from the UAVs flight telemetry (Gimbal, geolocation, velocity, altitude, attitude, etc.) is registered with data captured from the thermal camera to georeference image objects pixel location.

The nearest-neighbour method is regarded as the most straightforward approach to multimodal data registration [22]. This method attempts to associate a detection with the "closest" predicted position. The term "closest" is a predefined measurement of the distance between the object and the estimated position. However, as multi-objects begin to be tracked, nearest-neighbour is prone to becoming sub-optimal due to the order of associating measurements with tracked objects [22].

2.2.5 Visual SLAM for UAVs

Visual simultaneous localisation and mapping (SLAM) is a technique for estimating sensor motion and reconstructing structure in an unknown environment [30]. As UAVs require an accurate estimation of their state and can often not solely depend on Global Positioning System (GPS), visual SLAM has been employed in most UAV systems to provide vehicle pose in addition to a map of its environment [12]. Furthermore, research in visual-based SLAM methods used for UAV navigation tasks is increasing as small-scale autonomous aerial vehicles will play a significant role in the near future [33].

Visual SLAM techniques can be classified as stereo and monocular. The stereo method uses two or more cameras to process information, where the monocular only utilises a single camera. Initialising features using the monocular approach is a complex problem as it inherently does not have as much information as the stereo approach. A solution proposed by Davison et al. [9] uses a delayed initialisation algorithm that waits until the camera position has parallax enough to determine the position of the feature, where it can be included in the Kalman filter. A pose and map can be generated once enough features have been included.

Visual SLAM is an essential requirement for this project. As the UAV detects animals, these animals must be assigned a position on the virtual map that the SLAM algorithm produces, where tracks will be applied to predict where on the virtual map the animal should be at time x.

2.3 Contributions

As mentioned previously, this project extends Corcoran's automated detection of koalas using low-level aerial surveillance and machine learning, and Dr Denman's broadened use case in adapting Corcoran's work to detect different animals such as deer, improving the systems ability to track detections [8]. The tracking process moves from tracking objects in an image frame to a case where the system can track multiple objects positions and velocities in a world fixed coordinate frame. This process conversion enables tracked object positions to be estimated outside the UAV cameras field of view (FOV) for periods. Currently, the extended tracking process is appended to the end of the system, but it can be integrated where detections are made if real-time detection and tracking are required in future.

3 Methodology

The following subsections describe the process taken to track object detections. The order of sections is essential, as each subsequent step requires the conversions and calculations of its previous steps. Figure 4 below outlines a high-level overview of the tracking process.

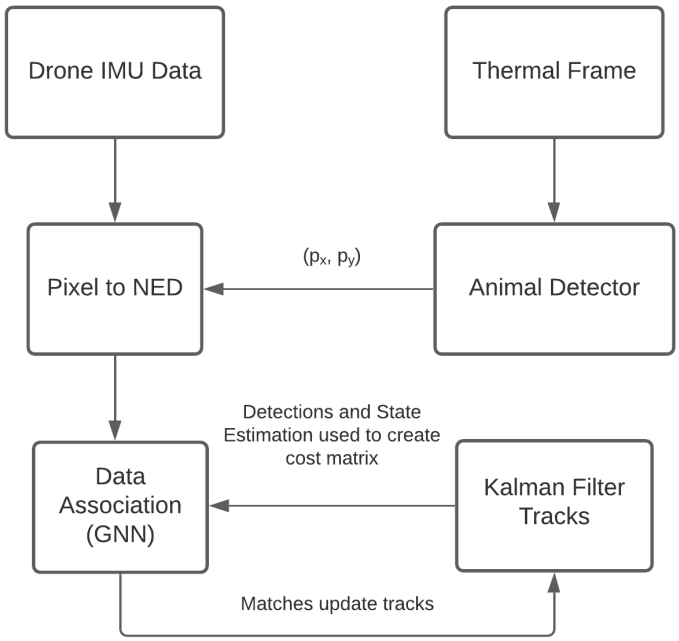


Figure 4: Animal Tracking Process

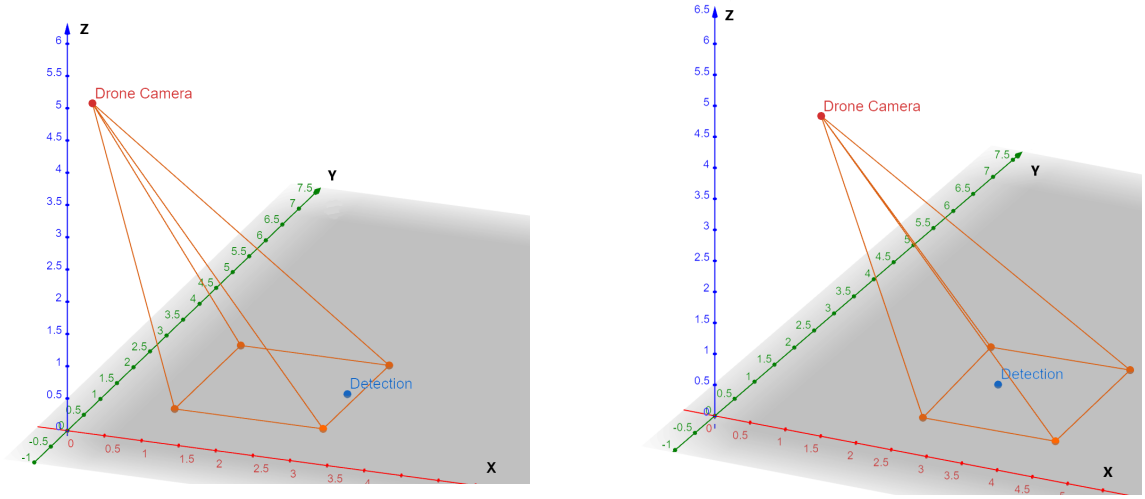
Prerequisites are detailed to ensure this system can be adapted and incorporated into other object detection algorithms.

3.1 Prerequisites

The following data variables are required to ensure the tracking process can be used for future UAV upgrades and work, aside from the object's location in the image frame returned by the object detector. In addition, the UAV's latitude (ϕ), longitude (λ) and height (h), as well as the cameras, roll, pitch and yaw are necessary for georeferencing transformation. This data is required for every frame of the thermal video and is typically embedded in a frames Exchangeable Image File (EXIF) data.

3.2 Pixel Location to NED Conversion

When an animal is detected in a thermal frame, the position of the detection is returned based on its location in the frame. For example, if an object was detected at the top middle of an angled camera frame, the coordinates $X = 320$ and $Y = 0$ are returned. This is an issue for the position, and velocity estimation as an object that appears at the top of the frame will move slower than one seen at the bottom, despite having the same velocity. Converting both observations to the same coordinate frame will ensure they have the same velocity when estimating their position: figures 5a and 5b below display a scenario where the UAV's camera frame detects a static object.



(a) Frame 1 *(b) Frame 2*

Figure 5: Demonstration of detected objects using NED coordinates across frames. The blue dot represents a detected object. Orange lines and points indicate the camera's projection and FOV.

The above example shows how a detected object moves across video frames. If the object were tracked only using its detected pixel location, frame one would return the object's location being $[320, 0]$, with frame 2 returning $[320, 350]$. Thus, although the object is not moving relative to the earth, it is moving relative to the camera frame. The goal of the georeference transformation is to maintain the object's static coordinates as the drone moves through space. The universal coordinate frame used in this project is the northeast-down (NED) coordinate system.

3.2.1 Earth-Centred, Earth-Fixed Frame

Before any transformation to NED, the UAV's lat, long and height are converted to an Earth-Centred, Earth-Fixed (ECEF) frame. The ECEF is a sound stage from the point of view of calculating orientations and directions for the UAV with global movements. ECEF takes a

set of Cartesian axes, X, Y and Z, with their origin at earth's centre [21]. Figure 6 below displays how the Cartesian axes are fixed.

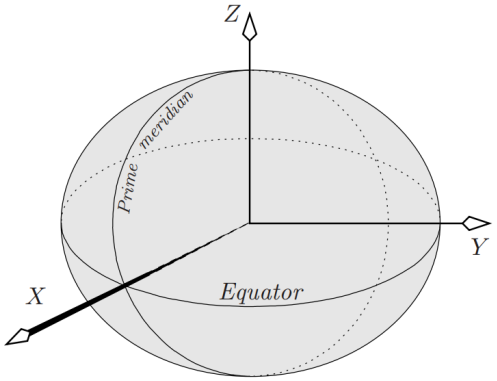


Figure 6: X-axis points to 0^o latitude, 0^o longitude. Y-axis points to 0^o latitude, 90^o longitude. Z-axis points to 90^o latitude, along Earth's axis of rotation [21].

As mentioned, the UAV's lat, long and height coordinates are converted to X, Y and Z as they are easier to work with. Thus, for any given lat-long-height points, X, Y and Z can be calculated by using Equations 1, 2 and 3 below.

$$X = (N(\phi) + h)\cos(\phi)\cos(\lambda) \tag{1}$$

$$Y = (N(\phi) + h)\cos(\phi)\sin(\lambda) \tag{2}$$

$$Z = (\frac{b^2}{a^2}N(\phi) + h)\sin(\phi) \tag{3}$$

where

$$N(\phi) = \frac{a^2}{\sqrt{a^2\cos^2(\phi) + b^2\sin^2(\phi)}} \tag{4}$$

Moreover, a and b are the equatorial radius and polar radius, respectively.

3.2.2 NED Coordinate Transformation

Once the coordinate position of the UAV is transformed to ECEF, a local geographic coordinate is created around the drone called NED. The NED coordinate system is a noninertial system with its origin fixed at the UAV's centre of gravity, with its coordinate frame fixed to the earth's surface [5]. Figure 7 shows the local directions of north, east and down respective to the UAV. Again, the curvature of the earth is neglected as the drone is operating over a small area.

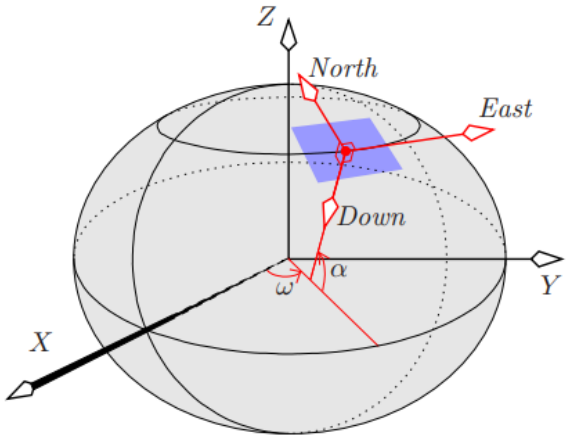


Figure 7: Local NED frame projection on earth's surface [21]

To convert between ECEF and NED coordinate frames, matrix rotations and transformations are required. Unlike traditional transformations from ECEF to NED, however, where the NED axes are relative to the UAV, the NED axes in this project must be relative to the camera frame. In addition, to aid in the simplification of calculations, it is assumed that the camera gimbal returns its rotation variables as if it were rotating around the earth's surface and not the UAV. If this step cannot be assumed and does rotate about the drone, a different rotation matrix must be added.

Conversion to NED starts by creating an initial set of NED axes on the equator and prime meridian. Each axe is represented by a unit vector in the ECEF frame, which all calculations are derived from:

$$N_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \qquad E_0 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \qquad D_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{5}$$

Each vector is rotated by its corresponding ECEF value. The first rotation is about the N_0 vector by X, E_0 vector by Y and D_0 by Z. The NED rotation is shown in Figure 8.

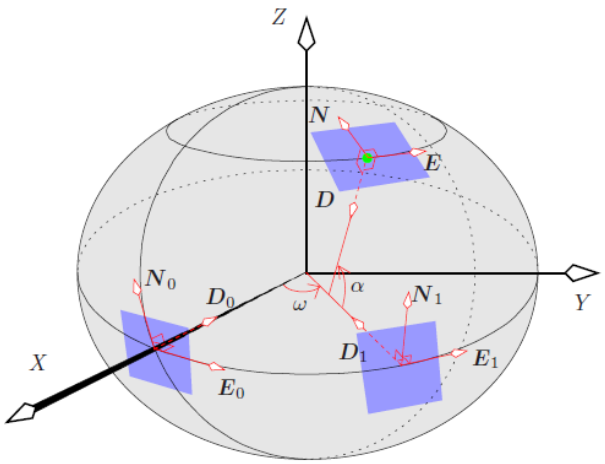


Figure 8: NED Axes Rotation [21]

As mentioned, however, a NED coordinate frame must be created around the camera frame, so object detections are all within the same coordinate plane. Therefore, further steps are required to project pixel locations onto the NED plane.

To simplify calculations, the pinhole camera model [24] is assumed. This assumption reduces the aperture of the camera's lens to zero, ensuring all light rays remain undeflected. Figure 9 demonstrates why this assumption is essential.

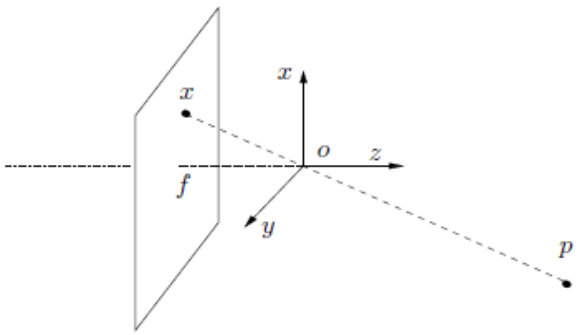


Figure 9: Pinhole Camera Model

Point p is the point x of the intersection of the ray going through the optical centre o and an image plane at a distance f from the optical centre [24]. From this, the NED coordinates can be calculated using the centroid of the detected objects pixel coordinates (p_{x_n}, p_{y_n}), using Equation 6.

$$\begin{bmatrix} x_n \\ y_n \\ 1 \end{bmatrix} = sKG \begin{bmatrix} p_{x_n} \\ p_{y_n} \\ 1 \end{bmatrix} \tag{6}$$

s is an arbitrary scaling factor because normalised homogeneous coordinates must be isolated. K and G are the intrinsic and extrinsic camera parameters. Finally, n is the current time step.

The intrinsic camera parameters, K, are required to link pixel coordinates of an image point with the corresponding coordinates in the camera reference frame. Thus, K is defined in Equation 7.

$$K = \begin{bmatrix} f\ s_x & 0 & c_x \\ 0 & f\ s_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{7}$$

f is the focal length of the camera lens, s_x and s_y refer to the pixel size of the sensor and c_x and c_y are the principal points in pixels. As we have assumed the pinhole camera model, the principal points are static and the frame's optical centre.

The extrinsic camera parameters, G, are defined by the rotation (R) and translation (T) matrices, shown in Equation 8, which calculate the rotation from the NED frame to the drones camera frames.

$$G_n = R_n T_n \tag{8}$$

The rotation matrix is represented as the rotation about the three individual gimbal axes; yaw (ψ), pitch (θ) and roll (ϕ). As mentioned previously, it is assumed that these axes are rotating around the flat Earth surface and not the drone itself. The rotation matrix is defined in Equation 9.

$$\begin{aligned} R_n &= \begin{bmatrix} \cos(\psi_n) & -\sin(\psi_n) & 0 \\ \sin(\psi_n) & \cos(\psi_n) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta_n) & 0 & \sin(\theta_n) \\ 0 & 1 & 0 \\ -\sin(\theta_n) & 0 & \cos(\theta_n) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi_n) & -\sin(\phi_n) \\ 0 & \sin(\phi_n) & \cos(\phi_n) \end{bmatrix} \\ &= \begin{bmatrix} r_{11_n} & r_{12_n} & r_{13_n} \\ r_{21_n} & r_{22_n} & r_{23_n} \\ r_{31_n} & r_{32_n} & r_{33_n} \end{bmatrix} \end{aligned} \tag{9}$$

The translation matrix, T, represents the drone's earth-centred, earth-fixed (ECEF) coordinates and is shown in Equation 10.

$$T_n = \begin{bmatrix} X_n \\ Y_n \\ Z_n \end{bmatrix} \tag{10}$$

The resulting extrinsic camera parameter equation is defined in Equation 11.

$$G_n = \begin{bmatrix} r_{11_n} & r_{12_n} & r_{13_n} \\ r_{21_n} & r_{22_n} & r_{23_n} \\ r_{31_n} & r_{32_n} & r_{33_n} \end{bmatrix} \begin{bmatrix} X_n \\ Y_n \\ Z_n \end{bmatrix} \tag{11}$$

The resulting x_n and y_n values is the object detected pixel location, converted to a north-east-down frame coordinate. These values will be used to estimate and predict the position of a detected object.

3.3 State Estimation

To assign previous observations to an appropriate new detection, it is helpful to predict where an object might be in the current frame. This problem is inherently difficult to solve as several hidden and unknown variables are associated with an animal's movement. The Kalman filter, proposed by Rudolf E. Kalman [19], attempts to estimate these hidden variables based on inaccurate and uncertain measurements. Estimating these variables provides the opportunity to predict where an object may be in the future.

A Kalman filter is comprised of two steps, prediction and correction [15]. The prediction step attempts to predict a previously detected objects state by applying Newton's motion equations:

$$x = x_0 + v_0 \Delta t + \frac{1}{2} a \Delta t^2 \tag{12}$$

where:

x = target's position
x_0 = target's initial position
v_0 = target's initial velocity
a = target's acceleration
Δt = time interval

With the assumption that the drone is tracking objects located on a flat surface and moving with a constant velocity, the following set of linear equations:

$$\begin{aligned} x_{n+1} &= x_n + V_{x,n} \Delta t + \frac{1}{2} A_{x,n} \Delta t^2 \\ y_{n+1} &= y_n + V_{y,n} \Delta t + \frac{1}{2} A_{y,n} \Delta t^2 \\ V_{x,n+1} &= V_{x,n} + A_{x,n} \Delta t \\ V_{y,n+1} &= V_{y,n} + A_{y,n} \Delta t \end{aligned} \tag{13}$$

where:

x_n = object position on x axis
y_n = object position on y axis
$V_{x,n}$ = objects x linear velocity
$V_{y,n}$ = objects y linear velocity
n = time step
Δt = time difference
$A_{x,n}$ = Gaussian white noise representing change in velocity in the x-direction
$A_{y,n}$ = Gaussian white noise representing change in velocity in the y-direction

This yields the following state extrapolation equation:

$$\begin{aligned} \hat{x}_{n+1} &= \mathbf{F}\hat{x}_n + \mathbf{B}\boldsymbol{u_n} \\ \hat{y}_n &= \mathbf{C}\hat{x}_n + \boldsymbol{v_n} \end{aligned} \tag{14}$$

where $\hat{x}_n$ is the predicted system state vector, $\begin{bmatrix} x_n, y_n, V_{x,n}, V_{y,n} \end{bmatrix}^T$, at time step n, $\hat{x}_n$ is an estimated system state vector and $\boldsymbol{u_n}$ is a control variable, $\begin{bmatrix} A_{x,n} & A_{y,n} \end{bmatrix}^T$, for the Gaussian white noise terms representing change in velocity. The $\boldsymbol{v_n}$ variable, $\begin{bmatrix} w_{x,n} & w_{y,n} \end{bmatrix}^T$, represents the Gaussian white noise for errors in object position measurements. $\mathbf{F}$, $\mathbf{B}$ and $\mathbf{C}$ are defined as the state transition, input transition and output matrices respectively, and are equal to:

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} \frac{1}{2}\Delta t^2 & 0 \\ 0 & \frac{1}{2}\Delta t^2 \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix}, \qquad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{15}$$

The Kalman filter attempts to calculate an estimate of $\hat{x}_n$ for an object's state at time step n using measurements of the position of the object, defined using $\hat{y}_n$ and predictions on the trajectory of the current object. It is important to note that to make a valid prediction of an object's future state, two subsequent frames with the same object must be collated to calculate the object's velocity. To perform the correction step of the Kalman filter, data association is required.

3.4 Data Association

In the field of multiple object tracking, data association is one of the most crucial parts. It enables the Kalman filter to become increasingly accurate at estimating an objects state as the filter corrects itself. Data association in the current system uses a similar GNN approach as seen in Konstantinova's [22] target tracking paper. As mentioned in Section 2.2.2, track updating involves two steps, *gating* and *association*.

3.4.1 Gating

Gating is a stern test that eliminates unlikely observations to track pairings. A 'gate' is formed around a predicted object position. All observations that fall within a gate threshold are considered for a tracking update. The distance from a predicted position to observation forms a gate and follows the Euclidean distance measurement defined in Equation 16.

$$d_{pq} = \sqrt{(q_x - p_x)^2 + (q_y - p_y)^2} \tag{16}$$

where:

p = previous observation predicted coordinates $\begin{bmatrix} x_{n+1} & y_{n+1} \end{bmatrix}$
q = Observed NED Coordinates

A gate threshold constant G is defined to determine if an observation should be considered for a tracking update. The observation will be appended to a list of other observations whose distance is less than G. G should be tuned to the arbitrary scaling factor s defined in Equation 6 above.

A two-dimensional Euclidean distance measurement is sufficient for this project due to the assumption that detected animals are on a flat surface. Konstantinova [22] implements a gating algorithm much more sophisticated than what is implemented; however, this is not required due to the assumptions and data present.

3.4.2 Association

Association is required when an observation falls within the gates of multiple target tracks or when multiple observations fall within the gate of a target track [22]. Optimal assignment of observations minimises the total summed distance for all individual assignments. Data association takes the output of the distance measurement, defined in Equation 16, and makes final measurement-to-track associations [4]. When a single observation is within the gate of a single track, a tracking update can be made immediately. A conflict scenario arises, where multiple observations fall within a single gate, or a single measurement falls within the gates of more than one track. Figure 10 demonstrates this complex assignment problem.

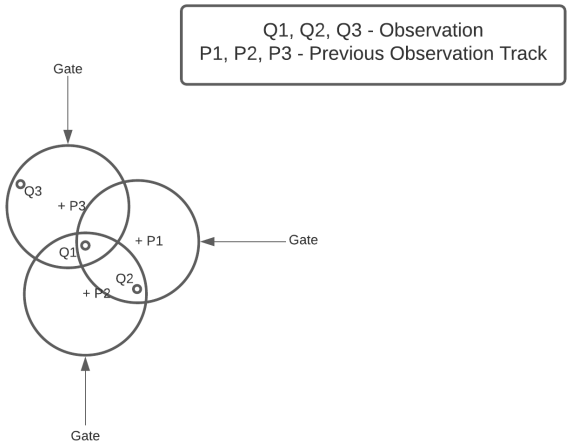


Figure 10: Conflict scenario where multiple observations fall within multiple gates

Optimal observation to track assignment is achieved using the Munkres assignment [27] algorithm. Assignments begin with the assumption that p tracks exist at the time of new observations, q. In the case of a cluttered environment, q does not always equal p. A cost matrix, defined in Equation 17 below, is built for the assignment problem solution defined.

$$C_{pq} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & \cdots & c_{1q} \\ c_{21} & c_{22} & c_{23} & \cdots & c_{2q} \\ c_{31} & c_{32} & c_{33} & \cdots & c_{3q} \\ \vdots & & & \ddots & \\ c_{p1} & c_{p2} & c_{p3} & \cdots & c_{pq} \end{bmatrix} \tag{17}$$

The elements of the cost matrix c_{pq} have the following values:

$$c_{pq} = \begin{cases} \text{inf}, & \text{if measurement } q \text{ IS NOT within the gate of track } p \\ d_{pq}, & \text{if measurement } q \text{ IS within the gate of track } p \end{cases}$$

The desired solution that the Munkres algorithm outputs minimise the summed total distance.

4 Data Capture

4.1 UAV

Before testing the performance of the described tracking algorithm, it is vital to understand how data is captured. As mentioned in Section 1, the DJI Matrice 210 V2 [11] was used to carry a Zenmuse XT 2 Flir [10] camera payload, shown in Figure 11. The setup used by the conservation team includes a second visual camera attached to a separate gimbal. This secondary camera was not utilised in this project. However, further developments could include its rich texture data to improve tracking further. The use of the secondary camera was investigated at the beginning of the project, however, unfortunately, technical complications arose, and the idea was abandoned. Section 8.1 below outlines an initial proposed solution.

Figure 11: DJI Matrice 210 V2 with Attached Gimbal

The Zenmuse XT 2 Flir camera, specific to this research, captures frames at a resolution of 640×512 at a rate of 8Hz. In addition, the attached camera lens used to capture data has a focal length of 13mm, with an image sensor pixel size of 17μm. These variables are the intrinsic camera parameters that define Equation 7.

Various test flights were analysed in this report, ranging in flight times. Most were flown over North Pine Dam and Rockhampton. A typical flight path is shown in Figure 12. The flight was conducted at North Pine Dam.

Figure 12: Drones path from Birds Eye View

As can be seen, the drone is programmed with a predetermined flight path, which fixes the drone's altitude at approximately 60 meters travelling in a 'lawn mower' pattern.

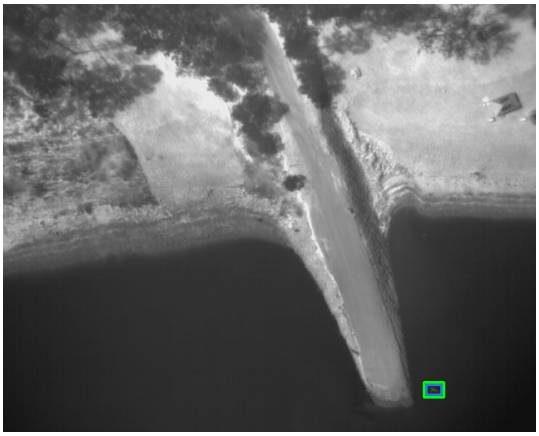
4.2 Frames

The Zenmuse XT 2 Flir camera captures thermal frames at eight frames per second (FPS), where the required metadata from the drone IMU is embedded in every eight frames of EXIF data. Upon completion of the flight, the output thermal video is passed through Cocoran's animal detection model. Although the detection model is accurate, missed and false detections still occur. Missed detections do not severely affect results as long as the animal is detected at some point. However, false detections do affect accuracy as a track will be assigned to the observation—figures 13 and 14 below outlines examples of missed and false detections observed across different test flights.

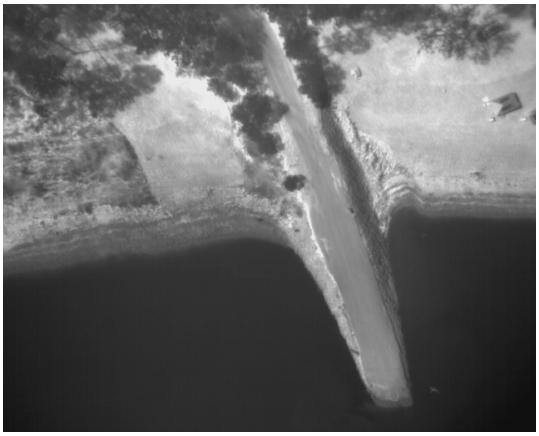
(a) True Detections

(b) Missed Detection Highlighted in Red Box

Figure 13: True Detection vs Subsequent Frame Missed Detection



(a) False Detection Highlighted in Green Box



(b) Original Image Frame of False Detection

Figure 14: False Detection Assumed as Unlikely Deer is in Water

To mitigate false detections, a degree of error will be applied to final tracking results. This will be a requirement regardless of false detections, as state estimation can be inaccurate over long periods.

5 Results

Due to a deficiency in the frequency of required data captured, resulting graphs will not display the significant movement of objects across frames. Regardless, this project does not aim to track observations across short periods but longer durations. To obtain results, the object detector confidence score was reduced to detect many animals. This, however, results in multiple types of animals being detected.

The results obtained are based on the final number of observation tracks identified compared against the number of detections. In addition, unique cases where an observation must be reidentified, or a track reassigned to the same animal in a cluster, will be outlined and described.

5.1 North Pine Dam - Flight 7 - 10/09/2019

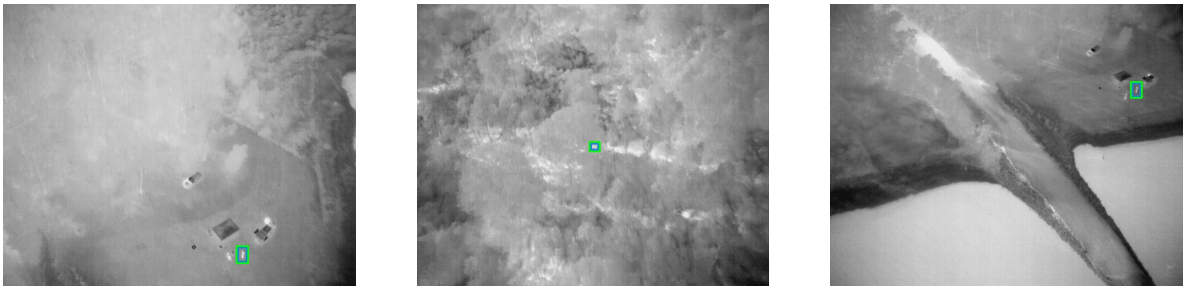
Flight 7 at North Pine Dam proved to be a good testing ground for the proposed developments as there were requirements for reidentification and multi-object tracking. Figure 15 below outlines the flight path the drone took.

Figure 15: North Pine Dam Flight Path - 10/09/2019

There were 30 detections across 324 observed frames for this flight, where 17 frames contained at least one detection. 13 unique animal tracks were identified after track analysis, with a ground truth of 13 tracks.

5.1.1 Reidentification

As mentioned, flight 7 outlined unique tracking scenarios this project addresses; the first is reidentification. Figures 16a and 16c display the scenario stated, with Figure 16b outlining an additional track observed between frames.



(a) Last Observation of Track 1 *(b) Track 2* *(c) Track 1 124 Frames Later*

Figure 16: Object Reidentification Scenario

After 264 frames, two individual observations have been identified. Table 1 below outlines the cost matrix generated for the observation identified in frame 265. The distance measurement listed is unitless and is the result of Equation 16.

	Distance
Track Number	Observation 1
1	9.2234
2	69.011

Table 1: Flight 7 Cost Matrix for Frame 265

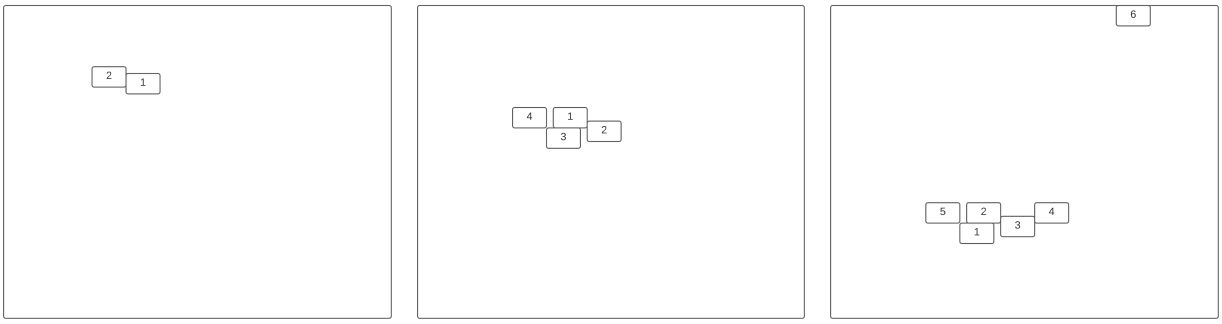
The shortest distance between the new observation coordinates and the predicted track position is 9.2234. With additional distance included for noise, this measurement provides a foundational gating threshold that subsequent gating assignments will obey. With a gating threshold in place, all frames can be processed.

5.1.2 Multi-Object Tracking

Figure 17 below demonstrates the requirement for data association. Frames 559 through 563 show the drone moving north (relative to the camera frame) and identifying new animals clustered together. This cluster grows in size as the thermal camera and object detector identify more animals. Based on the order of the output from the object detector, the rightmost identified animal is assigned track 1, with the leftmost assigned track 2. Figure 18 shows a simplified observation index assignment.

(a) Frame 559 *(b) Frame 561* *(c) Frame 563*

Figure 17: Multi-Object Scenario



(a) Frame 559 *(b) Frame 561* *(c) Frame 563*

Figure 18: Multi-Object Scenario Observation Assignment Index

The resulting cost matrix for frame 561 is shown in Table 2 below.

	Distance			
Track Number	Observation 1	Observation 2	Observation 3	Observation 4
1	9.451	10.000	11.134	9.840
2	9.907	10.700	11.647	9.969

Table 2: Multi-Object Detection Cost Matrix Frame 561

Minimising the distance function results in assigning observation 1 with track one and observation 4 with track 2. This is the correct assignment of observations to tracks.

The cost matrix derived from frame 563 is shown in Table 3

	Distance					
Track Number	Observation 1	Observation 2	Observation 3	Observation 4	Observation 5	Observation 6
1	18.520	15.801	17.111	17.723	16.093	19.516
2	18.979	16.240	17.785	18.637	16.218	20.331
3	17.784	15.111	16.210	16.619	15.667	19.504
4	16.764	14.048	15.345	15.983	14.393	21.000

Table 3: Multi-Object Detection Cost Matrix Frame 563

The minimum distance function assigns track 1 to observation 2, track 2 with observation 5, track 3 with observation four and track 4 to observation 3. This assignment is not entirely correct. Track 3 should be assigned to observation 3, with track four being assigned to observation 1. This misassignment results from measurement noise and will require finer tuning of the Kalman filter state estimation variables.

5.1.3 Tracks

Figure 19 below displays the detections made and their associated tracks.

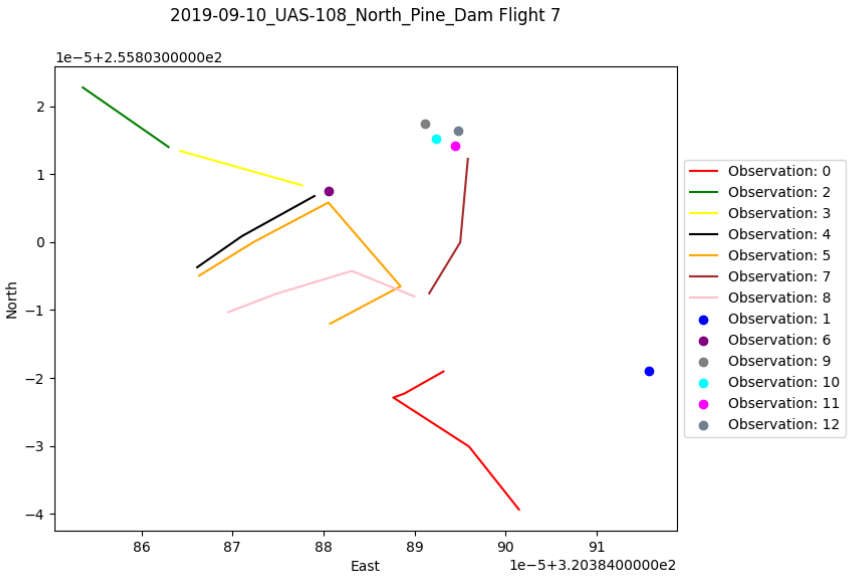


Figure 19: North Pine Dam Flight 7 Animal Tracks and Observations

The detections with a corresponding line demonstrate an animal that has been observed over subsequent frames, whose heading and velocity can begin to be estimated using state estimation. Single observed animals are represented with a dot. Due to the number of skipped frames, many animals were only detected once.

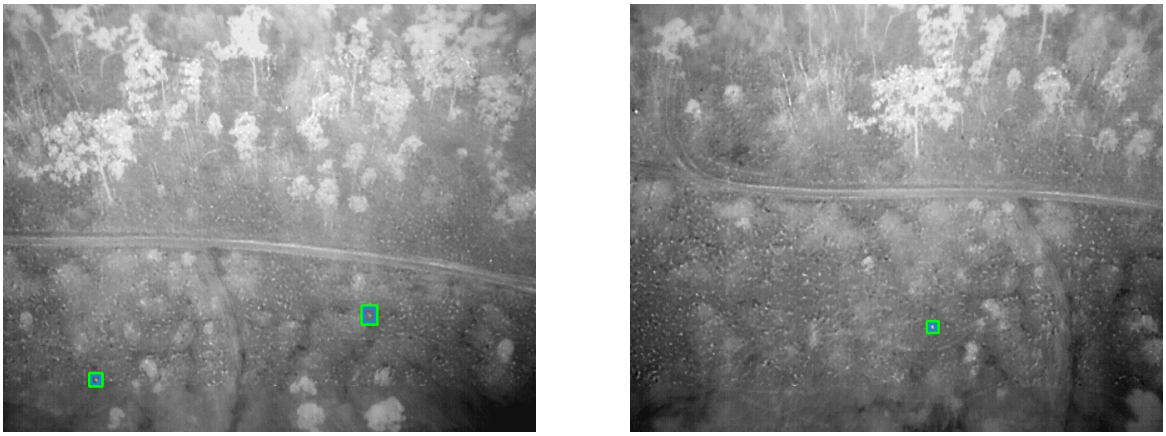
5.2 Rockhampton South - Flight 1 - 21/08/2019

Rockhampton South flight 1 was difficult to determine a ground truth as the flight path, shown in Figure 20, does not have significant points of reference, such as trees and roads, and frames are significantly noisier. Despite this, 25 unique animals were manually identified, where 20 animal tracks were returned after track analysis. This is reduced from 39 total detections across 428 frames.

Figure 20: Rockhampton South Flight Path - 21/08/2019

5.2.1 Reidentification

As per Section 5.1.1, we wish to determine a gating threshold by measuring the distance between a track and a particular associated new observation. Frame 21 below displays the same animal detected in two different frames. The fork in the dirt path has been used as a point of reference.



(a) Frame 349 *(b) Frame 361*

Figure 21: Same Observation Animal Observation

The animals identified in Frame 349 were assigned track 1 for the leftmost observation and track 2 for the rightmost. The resulting cost matrix for Frame 361 is shown in Table 4 below.

	Distance
Track Number	Observation 1
1	1.515
2	17.773

Table 4: Cost Matrix Frame 361

Correct assignment updates track 1 with the position of observation 1, with track two relying on the Kalman filter to predict its new position. The new observation distance defines a gating threshold of 5 for this flight.

5.2.2 Tracks

Figure 22 outlines the detections observed from Rockhampton South Flight 1.

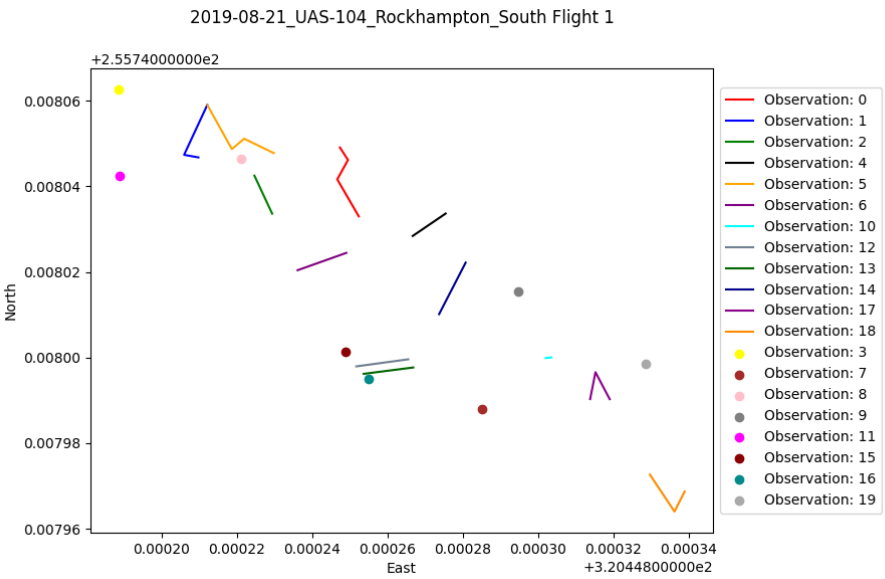


Figure 22: Rockhampton South Flight 1 Tracks

As shown, many observations were made throughout the test flight. However, due to measurement noise, the Kalman filter began to misinterpret an animals trajectory and velocity. This resulted in observation track states being estimated at an unreasonable distance from where they should be. In this case, these state estimations were disregarded, and the objects state remained static.

6 Discussion

The results outlined in Section 5 above demonstrate the correct functionality of the proposed tracking method, with some misassignments due to measurement noise. To resolve this, finer tuning of implemented Kalman filter is required. Through testing, it was found that a standard gating threshold cannot be used across flight tests—this required tuning of the threshold to maintain accuracy. A general gate threshold was determined that output was consistent across various flights. The following subsection discusses estimating a gating threshold.

6.1 Estimating General Gate Threshold

As the UAV's IMU is not calibrated the same for each flight, coordinate detection conversions will not be consistent. This poses an issue with hard coding a gating threshold. However, a general threshold can be determined through analysing several flights and comparing the ground truth number of detections with different gate threshold resulting detections. Table 5 below outlines the test conducted and the resulting ground truth verse different gate threshold tracks observed.

Flight	Total Observations	Ground Truth Tracks	Gate Threshold Distance						
			1	3	5	7	10	15	20
1	39	25	39	38	35	31	28	20	16
2	30	20	28	27	25	24	24	18	16
3	101	34	99	90	74	67	49	37	18
4	4	3	4	4	3	3	3	3	2
5	10	8	10	9	9	9	8	6	6
6	42	8	42	40	33	22	15	8	8

Table 5: Gate Thresholds vs Ground Truth

As shown, the gating threshold with the best performance across various flights is 15. It has shown consistent results across multiple flights using the discussed camera and drone setup.

7 Conclusion

This project sought to improve wildlife abundance monitoring by building on the work of Cochran et al. This was achieved by first transforming and projecting pixel detection coordinates onto a world coordinate frame, ensuring a detection seen at frame 0 will be on the same plane as a detection seen at frame 100. Projection onto a world coordinate frame is essential to estimate an objects state to enable reidentification. A Kalman filter was utilised to predict an objects state. Once an object leaves the camera FOV, it is crucial to estimate where it may be in future frames if it is detected again. As IMU and detection data has inherent noise associated, the Kalman filter was tuned to manage. A GNN method was implemented to handle MOT. The GNN method comprises two steps, gating and association. Gating eliminates unlikely objects to track pairings where association minimises the total distance between objects and tracks using the Munkres algorithm.

7.1 Future Work

Although this project was a success, more work is required to improve the consistency of results. The following subsections outline improvements that can be made in the future.

7.1.1 Kalman Filter

The current state estimation has several issues associated. First, measurement noise produced by the FLIR camera and IMU data was not entirely understood, leading to misclassifications due to wrong state predictions. Second, further work is required to improve the Kalman filter. Although it is not required to introduce an extended Kalman filter [18] as we are assuming linear movement, improvements to the measurement, measurement uncertainty, process noise uncertainty and estimation uncertainty equations are essential.

7.1.2 Data Quality

Improvements to data density will be required for future work. As mentioned in Section 4.2, few images have the required EXIF metadata outlined in the prerequisites defined in Section 3.1. Embedding the required EXIF data in all captured thermal frames or exporting the data concerning its correlated frame will solve this issue. Unfortunately, this was not possible for this project as the test flights analysed were prerecorded. Another possible solution for the absence of data is interpolating extrinsic camera and location data between missing EXIF embedded frames. This will impact accuracy, however.

7.1.3 Different Cameras

One goal of this project was to enable it to be adapted and carried over to another dynamic detection system and work seamlessly with minor alterations, as long as the prerequisites mentioned in Section 3.1 are satisfied. The only alterations required are the intrinsic camera parameters that from Equation 7. To ensure simple adaption, testing will be conducted using different types of cameras attached to the drone. If successful, arguments can be added to the code base where the camera parameters can be altered.

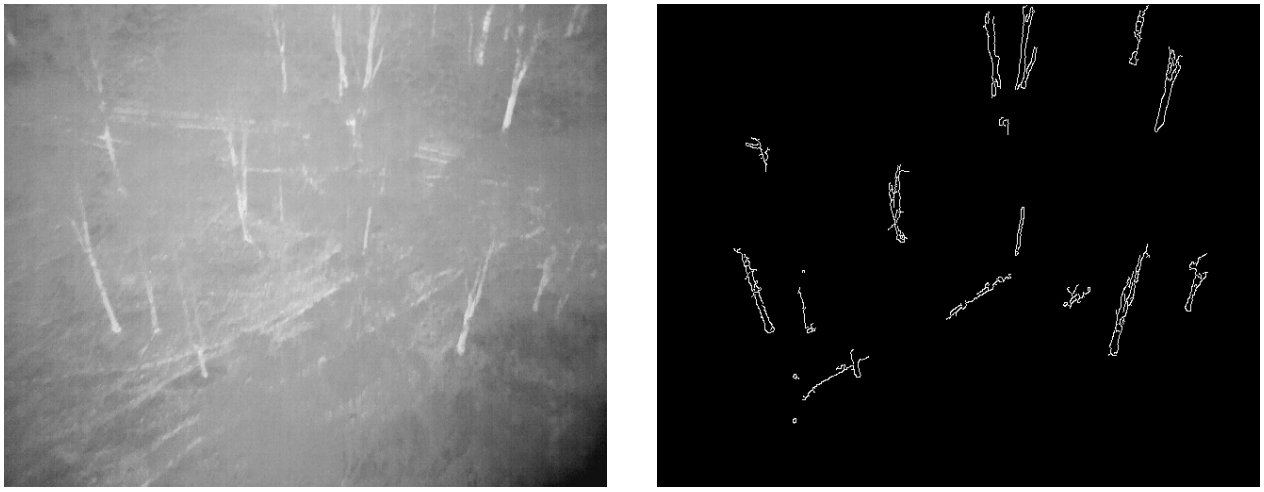
8 Timeline

As mentioned in Section 4.1, an effort was made to utilise thermal and visual cameras to capture more rich data that would aid in tracking. The following subsection outlines the efforts made before the decision was made to abandon the idea.

8.1 Previous Work

An efficacious method to improve Evangeline Cocorcan's current automated detection of koalas is proposed. Improving detection and camera tracking will be achieved through information and sensor fusion of thermal and visual data captured by a Zenmuse XT2 mounted to a DJI Matrice 210 V2. The inherent properties of a thermal camera suit the use case of enabling an ML model to identify animals on the ground or in trees as they generally emit more heat than their surrounding environment. However, due to the surrounding environment having a similar heat signature, it is challenging to track camera view changes; as a result, there not being many significant points of reference to track as it lacks texture information. The inverse of a thermal camera for this use case is a visual camera. Visual data contains rich texture information that can help estimate changes in camera position and thus help track detections as a drone moves; however, detecting animals within this data alone is challenging. Again, this is due to animals simply blending into their surrounding environment.

Utilising and fusing both thermal and visual data can significantly improve detection and camera tracking. A method has been developed to template match and aligns equivalent thermal and visual frames together. To achieve this, a thermal frame is loaded, converted to greyscale, where its edges are detected using OpenCV's Canny edge detection algorithm. This conversion is shown in Figures 23a and 23b below.



(a) Original Thermal Image *(b) Edged Thermal Image*

Figure 23: Converted Thermal Image

The original visual image is then converted to greyscale, resized and run through the same Canny edge detection algorithm. Finally, template matching is performed to search and find the location of a template image (edged thermal image) in a larger image (scaled edged visual image). The result is a visual image that has been cropped to the relative thermal image position and size. The resulting cropped and aligned image is shown in Figure 24b below, with the thermal reference image shown in Figure 24a.

(a) Original Thermal Image *(b) Cropped Visual Image*

Figure 24: Cropped Visual Image Comparison

Due to differences in the field of view, the images are not precisely aligned. This is a problem that will need to be solved to continue forward. Another step is aligning frames across both thermal and visual video sources. According to the Zenmus XT2 specifications, the thermal camera exports its video feed at nine frames per second (FPS), whereas the visual

camera exports at 30 fps. As we will preserve as much information as possible, investigations surrounding aligning frames across mismatched frame timings must be undertaken.

The ultimate end goal of this project seeks to build on the current process above and develop methods to register the visual and thermal streams and exploit the aligned data to improve the performance of animal detection. This will be achieved using camera and drone telemetry to improve localisation and utilising visual data to strengthen re-detection.

References

[1] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua, "Multiple object tracking using k-shortest paths optimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 9, pp. 1806–1819, 2011. DOI: 10.1109/TPAMI.2011.21.

[2] A. J. Bewley, "Vision based detection and tracking in dynamic environments with minimal supervision," Ph.D. dissertation, Queensland University of Technology, 2018. DOI: 10.5204/thesis.eprints.116014. [Online]. Available: https://eprints.qut.edu.au/116014/.

[3] S. Blackman, *Design and analysis of modern tracking systems*. Boston: Artech House, 1999, ISBN: 978-1580530064.

[4] P. Bogler, "Radar principles with applications to tracking systems," 1990.

[5] G. Cai, B. M. Chen, and T. H. Lee, "Coordinate systems and transformations," in *Unmanned Rotorcraft Systems*. London: Springer London, 2011, pp. 23–34, ISBN: 978-0-85729-635-1. DOI: 10.1007/978-0-85729-635-1_2. [Online]. Available: https://doi.org/10.1007/978-0-85729-635-1_2.

[6] J. Candamo, M. Shreve, D. B. Goldgof, D. B. Sapper, and R. Kasturi, "Understanding transit scenes: A survey on human behavior-recognition algorithms," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 1, pp. 206–224, 2010. DOI: 10.1109/TITS.2009.2030963.

[7] D. Cooper, "Multiple target tracking with radar applications," *Electronics and Power*, vol. 33, no. 6, p. 407, 1987. DOI: 10.1049/ep.1987.0251. [Online]. Available: https://doi.org/10.1049/ep.1987.0251.

[8] E. Corcoran, S. Denman, J. Hanger, B. Wilson, and G. Hamilton, "Automated detection of koalas using low-level aerial surveillance and machine learning," *Scientific Reports*, vol. 9, p. 3208, 2019. DOI: https://doi.org/10.1038/s41598-019-39917-5.

[9] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, 2007. DOI: 10.1109/TPAMI.2007.1049.

[10] DJI, *Zenmuse xt 2*, English, version Version 1.0, Apr. 2018, 18 pp.

[11] ——, *Matrice 200 series v2*, English, version Version 2.0, Jul. 2020, 79 pp.

[12] G. Fink, M. Franke, A. F. Lynch, K. Röbenack, and B. Godbolt, "Visual inertial slam: Application to unmanned aerial vehicles," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 1965–1970, 2017, 20th IFAC World Congress, ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2017.08.162. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896317302859.

[13] P. Fretwell, I. Staniland, and J. Forcada, "Whales from space: Counting southern right whales by satellite," *PloS one*, vol. 9, e88655, Feb. 2014. DOI: 10.1371/journal.pone.0088655.

[14] P. Gabrlik, "Transformation of uav attitude and position for use in direct georeferencing," Jan. 2016.

[15] M. Grewal, *Kalman filtering : theory and practice using MATLAB*. Hoboken, New Jersey: Wiley, 2015, ISBN: 9781118984987.

[16] T. Hollings, M. Burgman, M. van Andel, M. Gilbert, T. Robinson, and A. Robinson, "How do you find the green sheep? a critical review of the use of remotely sensed imagery to detect and count animals," *Methods in Ecology and Evolution*, vol. 9, no. 4, pp. 881–892, 2018. DOI: https://doi.org/10.1111/2041-210X.12973. eprint: https://besjournals.onlinelibrary.wiley.com/doi/pdf/10.1111/2041-210X.12973. [Online]. Available: https://besjournals.onlinelibrary.wiley.com/doi/abs/10.1111/2041-210X.12973.

[17] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 34, no. 3, pp. 334–352, 2004. DOI: 10.1109/TSMCC.2004.829274.

[18] S. J. Julier and J. K. Uhlmann, "New extension of the kalman filter to nonlinear systems," in *Defense, Security, and Sensing*, 1997.

[19] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, Mar. 1960, ISSN: 0021-9223. DOI: 10.1115/1.3662552. eprint: https://asmedigitalcollection.asme.org/fluidsengineering/article-pdf/82/1/35/5518977/35_1.pdf. [Online]. Available: https://doi.org/10.1115/1.3662552.

[20] I. S. Kim, H. S. Choi, K. M. Yi, J. Y. Choi, and S. G. Kong, "Intelligent visual surveillance — a survey," *International Journal of Control, Automation and Systems*, vol. 8, no. 5, pp. 926–939, Oct. 2010. DOI: 10.1007/s12555-010-0501-4. [Online]. Available: https://doi.org/10.1007/s12555-010-0501-4.

[21] D. Koks, "Using rotations to build aerospace coordinate systems," Australian Government - Department of Defence, Edinburgh, SA 5111, Australia, Tech. Rep. DSTO–TN–0640, Jul. 2008.

[22] P. Konstantinova, A. Udvarev, and T. Semerdjiev, "A study of a target tracking algorithm using global nearest neighbor approach," Jan. 2003. DOI: 10.1145/973620.973668.

[23] F. S. Leira, H. H. Helgesen, T. A. Johansen, and T. I. Fossen, "Object detection, recognition, and tracking from uavs using a thermal camera," *Journal of Field Robotics*, vol. 38, no. 2, pp. 242–267, 2021. DOI: https://doi.org/10.1002/rob.21985. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.21985. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21985.

[24] Y. Ma, S. Soatto, J. Košecká, and S. S. Sastry, *An Invitation to 3-D Vision*. Springer New York, 2004. DOI: 10.1007/978-0-387-21779-6. [Online]. Available: https://doi.org/10.1007/978-0-387-21779-6.

[25] E. Maggio and A. Cavallaro, "Learning scene context for multiple object tracking," *IEEE Transactions on Image Processing*, vol. 18, no. 8, pp. 1873–1884, 2009. DOI: 10.1109/TIP.2009.2019934.

[26] C. R. McMahon, H. Howe, J. van den Hoff, R. Alderman, H. Brolsma, and M. A. Hindell, "Satellites, the all-seeing eyes in the sky: Counting elephant seals from space," *PLoS ONE*, vol. 9, no. 3, Y. Ropert-Coudert, Ed., e92613, Mar. 2014. DOI: 10.1371/journal.pone.0092613. [Online]. Available: https://doi.org/10.1371/journal.pone.0092613.

[27] J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of The Society for Industrial and Applied Mathematics*, vol. 10, pp. 196–210, 1957.

[28] L. L. PATER, T. G. GRUBB, and D. K. DELANEY, "Recommendations for improved assessment of noise impacts on wildlife," *The Journal of Wildlife Management*, vol. 73, no. 5, pp. 788–795, 2009. DOI: https://doi.org/10.2193/2006-235. eprint: https://wildlife.onlinelibrary.wiley.com/doi/pdf/10.2193/2006-235. [Online]. Available: https://wildlife.onlinelibrary.wiley.com/doi/abs/10.2193/2006-235.

[29] A. C. Seymour, J. Dale, M. Hammill, P. N. Halpin, and D. W. Johnston, "Automated detection and enumeration of marine wildlife using unmanned aircraft systems (uas) and thermal imagery," *Scientific Reports*, vol. 7, no. 1, p. 45 127, 2017. DOI: 10.1038/srep45127.

[30] T. Taketomi, H. Uchiyama, and S. Ikeda, "Visual SLAM algorithms: A survey from 2010 to 2016," *IPSJ Transactions on Computer Vision and Applications*, vol. 9, no. 1, Jun. 2017. DOI: 10.1186/s41074-017-0027-2. [Online]. Available: https://doi.org/10.1186/s41074-017-0027-2.

[31] S. Wang, L. Ding, Z. Chen, and A. Dou, "A rapid uav image georeference algorithm developed for emergency response," *Journal of Sensors*, vol. 2018, pp. 1–10, Oct. 2018. DOI: 10.1155/2018/8617843.

[32] X. Wang, "Intelligent multi-camera video surveillance: A review," *Pattern Recognition Letters*, vol. 34, no. 1, pp. 3–19, 2013, Extracting Semantics from Multi-Spectrum Video, ISSN: 0167-8655. DOI: https://doi.org/10.1016/j.patrec.2012.07.005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S016786551200219X.

[33] S. Weiss, D. Scaramuzza, and R. Siegwart, "Monocular-slam–based navigation for autonomous micro helicopters in gps-denied environments," *Journal of Field Robotics*, vol. 28, no. 6, pp. 854–874, 2011. DOI: https://doi.org/10.1002/rob.20412. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.20412. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.20412.

[34] B. Zhan, D. N. Monekosso, P. Remagnino, S. A. Velastin, and L.-Q. Xu, "Crowd analysis: A survey," *Machine Vision and Applications*, vol. 19, no. 5-6, pp. 345–357, Apr. 2008. DOI: 10.1007/s00138-008-0132-4. [Online]. Available: https://doi.org/10.1007/s00138-008-0132-4.

[35] C. Zhang and J. M. Kovacs, "The application of small unmanned aerial systems for precision agriculture: A review," *Precision Agriculture*, vol. 13, no. 6, pp. 693–712, 2012. DOI: 10.1007/s11119-012-9274-5.

[36] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212–3232, 2019. DOI: 10.1109/TNNLS.2018.2876865.

A Flight Path with Elevation

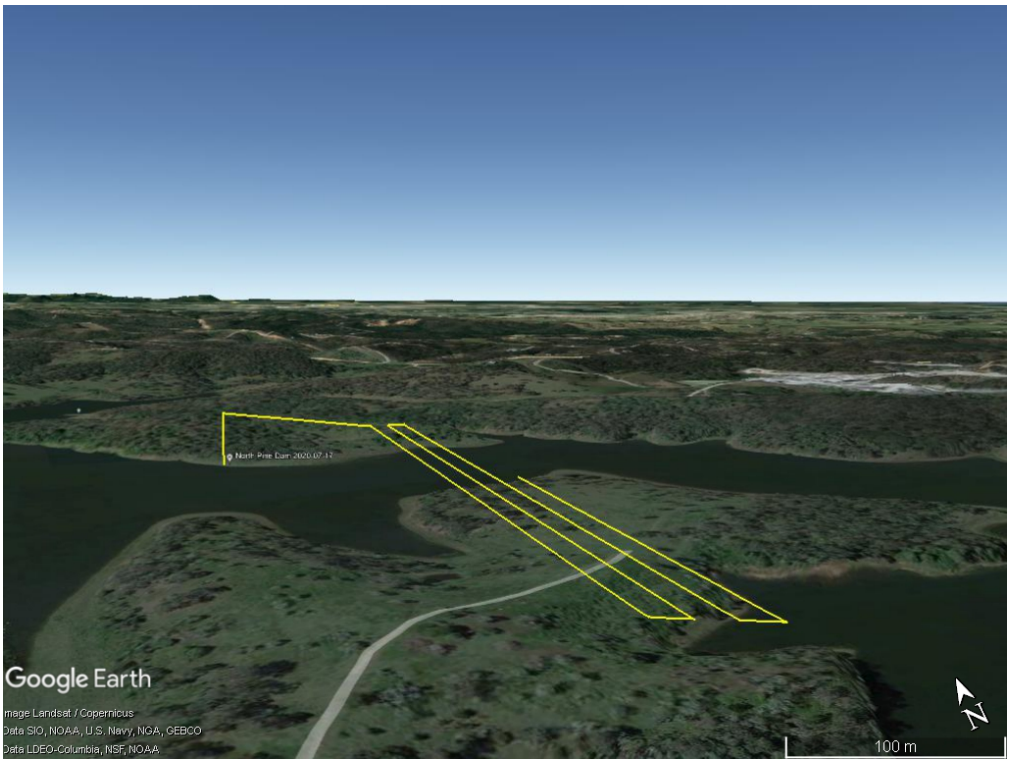


Figure 25: Drones Path with Elevation