Camera Lidar Fusion for Unmanned Aerial Vehicle Detection

Cosimo Leuci[1], Sedat Dogru[2] and Lino Marques[2]
[1]Department of Electronics and Telecommunications (DET), Polytechnic University of Turin,
10129 Turin, Italy s317823@studenti.polito.it
[2]Institute of Systems and Robotics, Department of Electrical and Computer Engineering,
University of Coimbra, 3030-290 Coimbra, Portugal {sedat, lino}@isr.uc.pt

***Abstract*—Due to their low-cost and ease of use, cameras are very popular for drone detection. Thanks to the feature reach images obtained from the cameras, it is possible to classify drones with a relatively high success ratio. However, cameras lack the means for accurate localization, allowing only a rough measure of the orientation of the target UAV. In order to better localize the UAV, in this study we fuse camera and lidar measurements, localizing UAVs accurately. This fusion was implemented using a Kalman filter, which enables integration of both sensor data, improving detection and tracking accuracy**

***Index Terms*—UAV, drone, lidar, detection, camera, fusion**

I. Introduction

Applications of Unmanned Aerial Vehicles (UAV), which are colloquially known as drones, are numerous and range from agriculture and environmental monitoring to security and logistics [1], [2]. Their role in the world is becoming more and more important in helping people but, unfortunately, they are also playing an increasing role in targeting civilians, surveillance abuse and disturbing air-travel. This improper use of UAVs has triggered research on their detection, resulting in different detection systems using various sensors, such as cameras [3]–[6], lidar [7]–[9] and radar [10], [11].

Using 3D lidars, such as the Velodyne VLP-16, presents some difficulties with increasing distance. Although they can construct a dense point cloud of a nearby drone, at far distances the detection often consists of only one or two data points [7], and only when the drone is directly along one of the scanning beams, not between them. Hence, making tracking and classification very difficult. Despite these limits, lidar is a valuable tool for detecting drones at considerable distances, offering 3D coordinates that are not readily accessible through alternative drone detection technologies.

Detection and tracking are also performed using camera sensors, but object detection is still difficult in videos captured by a moving camera, because camera motion and object motion are mixed [12]. An open issue in dynamic background subtraction is that the images may contain some elements that are not completely static, such as water surface or waving trees. Therefore, it is possible to find false clusters in the images [13]. In order to detect and track using moving cameras, optical flow focusing on the different displacement of the target in the image sequence has also been used [14]. On the other hand, cameras allow relatively easier object detection and classification when the objects are sufficiently visible.

In this work we propose a new method for drone detection, a method based on fusion of camera and lidar data. The method uses the camera for detection, followed by the lidar for full 3D localization, and then followed by the camera again for subsequent tracking. The method switches between the camera and the lidar depending on how the tracking errors evolve and the target drone appears. The approach, utilizing the strength of each sensor, allows more robust drone detection and tracking.

II. Related Work

Camera-based object detection has been researched extensively in the literature. Chapel and Bouwmans [13] elaborated an important review analyzing the main challenges of detection using a moving camera. In the literature, background subtraction is a common method used in the pipeline of object detection with a camera. However the process becomes more complicated when used for images taken by moving cameras. A low frame rate often does not allow for a clear distinction between background and foreground changes. Hu et al. [12], using a moving camera, proposed a method that detects the feature points in the frames using a modified Harris corner detector [15], and that then classifies them as belonging to foreground or background. Jiang et al. [16] proposed multi-camera 3D-object detection for autonomous driving using an onboard camera from the ego car's perspective. Birch and Woo [5] conducted an evaluation of the visual detection of drones at various wavelengths within the visible and infrared spectra, finding that visual band analysis is more efficient with a uniform background.

Several approaches use lidar to detect drones. Hammer et al. [17] investigated the feasibility of drone detection and they conducted experiments using a platform equipped with two Velodyne VLP-16 and two Velodyne HDL-64 lidars. Sier

This work was financially supported by the AM2R project "Mobilizing Agenda for business innovation in the Two Wheels sector" funded by PRR - Recovery and Resilience Plan and by the Next Generation EU Funds, under reference C644866475-00000012 | 7253.

This work was performed during Erasmus stay of C. Leuci at the University of Coimbra.

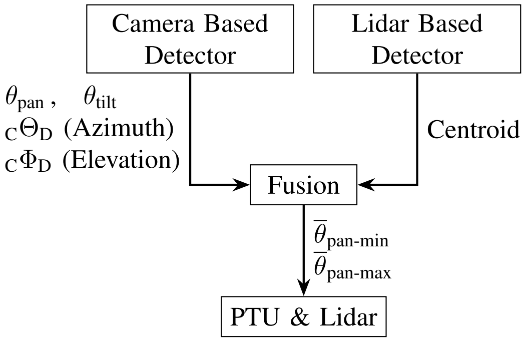

Camera Based Detector
Lidar Based Detector
θ_{pan}, θ_{tilt}
${}_{\text{C}}\Theta_{\text{D}}$ (Azimuth)
${}_{\text{C}}\Phi_{\text{D}}$ (Elevation)
Centroid
Fusion
$\overline{\theta}_{\text{pan-min}}$
$\overline{\theta}_{\text{pan-max}}$
PTU & Lidar

Fig. 1: Workflow of the detection and tracking process

et al. [9] used lidar-generated images for real-time tracking, fusing the lidar-images and the point clouds from a single lidar sensor for tracking UAVs. Catalano et al. [8] used a ground-based UAV tracking system utilizing a solid-state lidar with adjustable frequency and varying FoV (Field of View) coverage, proposing a method in which the lidar frame integration time is dynamically adjusted based on the distance and speed of the UAV, obtaining a better accuracy and persistence. Dogru and Marques [7] studied sparse detection with the aim of not only detection but also motion estimation and active tracking of the target drones. They calculated the probability of detection under different scenarios for a lidar-turret system, and introduced a tracking approach to allow continuous tracking of the detected drones.

Camera and lidar fusion has been tested for collision avoidance systems, providing more accurate and robust detections for overlapping objects [18]. Caltagirone et al. [19] fused camera images and lidar point clouds for carrying out road detection using a fully convolutional neural network (FCN). Liu et al. [20] proposed a new deep neural network (namely FuDNN), that improves the performance of lidar-camera fusion for autonomous driving.

Although sensor fusion of lidar and cameras is common in the literature with examples on vehicle detection [21], we are not aware of any study fusing the two for UAV detection. This work focuses on the fusion of lidar and camera for UAV detection. Lidars, thanks to their shorter wavelengths and precise laser scanning capabilities, allow accurate measurement of the distances to targets in three-dimensional space. Cameras allow continuous scanning of the environment in a relatively cheap manner, helping also in the identification of the drones in the images. However, they lack accurate distance measurement.

III. Method

The flow-chart involves three principle and one secondary blocks that interact with each other (Fig. 1). The principle blocks are responsible for detecting the drone in the images obtained from the camera, for detecting the drone using point cloud data obtained from a 3D lidar, and fusing them in the Fusion block. The secondary block (PTU & Lidar) is responsible for controlling the pan-tilt units holding the lidar and the camera. The *Camera Based Detector* takes the rectified image as input, whereas the *Lidar Based Detector* takes the point cloud data as input. Initially, both blocks save the respective backgrounds to later run background subtraction. Later, after background subtraction, they find the coordinates of the centroids of the objects. The *Fusion* block receives the data, performs fusion, and generates the data necessary to track the drone's path in 3D space.

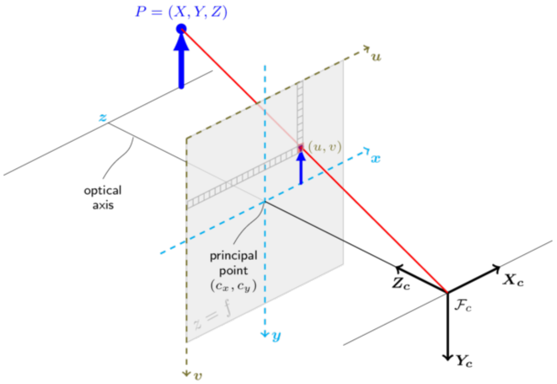


Fig. 2: The pinhole camera model

A. Camera Based Drone Detection

Camera based drone detection aims to detect the drone in a given image, as well as its 3D orientation with respect to the camera. The orientation contains the azimuth and elevation angles of the detected drone. This block first receives the images in RGB format and then it converts them to the HSV format. HSV is a common cylindrical-coordinate representation of color images, and it is used in image analysis and computer vision. In this color space, which roughly separates color from light, thresholding is relatively simpler, allowing easy creation of relatively more robust binary masks. The block subtracts the new images from the first ones, which serve as the background images. Then a thresholding in the V of the HSV domain is applied. Subsequently erosion operation is applied, removing sparse white pixels, thereby making the calculation of the centroids more accurate. The value for the threshold was found manually before each flight, because it was observed to vary throughout the day due to variations in light. However, analyzing many frames 30 % threshold for the third channel of HSV was found to allow perfect detection of the drone in flights summarized in this paper.

After finding the centroid in the image, we used the pinhole camera model to estimate the angles of the drone with respect to the camera (Fig. 2). This model can be used as a first approximation of the mapping between a 3D scene and its corresponding 2D image. The following equations are satisfied for the image of a 3D point on the camera in the pinhole camera model [22]:

$$u = f_x * x' + c_x \tag{1}$$
$$v = f_y * y' + c_y \tag{2}$$

where x' and y' represent respectively the ratios $\frac{x}{z}$ and $\frac{y}{z}$. The parameters f_x, f_y, c_x, c_y are known as the intrinsic camera parameters, with the first two being the focal length and the later two being the principal point of the image. The camera parameters are usually estimated following a calibration sequence, which we performed initially, before any flight. The parameters are usually presented in the camera matrix A as

$$A = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \tag{3}$$

Using the above identities the azimuth (${}_{\mathrm{C}}\Theta_{\mathrm{D}}$) and elevation (${}_{\mathrm{C}}\Phi_{\mathrm{D}}$) angles of the drone with respect to the camera can be found using

$${}_{\mathrm{C}}\Theta_{\mathrm{D}} = \arctan\left(\frac{x}{z}\right) \tag{4}$$

$${}_{\mathrm{C}}\Phi_{\mathrm{D}} = \arctan\left(\frac{y}{z}\right) \tag{5}$$

In this work, the camera was placed on top of a pan tilt unit (PTU) to allow easy positioning of the camera. Hence, the angles obtained above are also transformed to the base of the PTU to have consistent angles with respect to the lidar's PTU, as to be explained in the next section.

B. Lidar Based Drone Detection

The lidar is placed on top of a PTU to increase the field of view. On start, the lidar-based detection block performs a continuous sweep of the environment for 15 s to construct a point cloud corresponding to the background. After the initial background collection, the PTU is rotated to an initial angle and left waiting for detections to be reported by the camera node. After receiving an alarm from the camera, the lidar-PTU is rotated to align the lidar with the reported angle of the target drone, and processing of the point cloud data begins. First, an x, y, z filter that removes points through simple thresholding is applied, filtering the point cloud roughly. Afterwards, background subtraction is applied, removing all points corresponding to pre-existing objects, such as parts of the trees and parts of the ground. Then the points of the cloud around the angles reported by the camera are found, they are clustered and their centroids are reported. Filtering the pointcloud is necessary to eliminate pre-existing objects that may have similar angular distances to the drone with respect to the lidar.

The background is constructed as a voxel grid of the collected point cloud data. The voxel grid helps both smooth the point cloud and improve the search speed when comparing the new pointclouds to the background. The voxel grid is represented as a k-d tree (k-dimensional tree) data structure to organize points in k-dimensional space. This structure is ideal for range and nearest neighbor searches, focusing on three-dimensional point clouds.

Point cloud clustering after filtering is done using Euclidean distance, clustering the points based on several criteria: maximum distance between two points in a cluster, minimum and maximum number of points in a cluster. Both in background construction and clustering we made use of the well known Point Cloud Library[1] (PCL). After clustering, the coordinates of the centroid of the object are sent to the fusion block.

C. Camera and Lidar Fusion

The fusion block receives the centroid of the detected object and its angles from the camera, and then sends these angles to control the PTU of the lidar. For fusion a hybrid Extended Kalman Filter (EKF) was used. A constant speed model was used for the drone. A linear system was assumed for the lidar update, in which the measurements are the 3D Cartesian coordinates of the centroid. A non-linear system was assumed for the camera update, in which the measurements were the azimuth and elevation angles of the drone with respect to the world frame.

The fusion block also monitors the workflow of drone detection. It uses different states for the camera and the lidar. The states are defined as NO DETECTION, INITIALIZATION, LIDAR TRACKING and CAMERA TRACKING. The purpose of the work is to detect the drone with the camera and send the angular position of the drone to the lidar's PTU, so that the lidar can start to detect and to track the drone. Afterwards the lidar is expected to allow the camera to keep tracking until the target changes its maneuver, or the EKF has a large covariance in the state estimate. These states change depending on the inputs of the block. Firstly the state is initialized to NO DETECTION, that means the camera and lidar are saving the background and they have not detected anything. Then when an object is detected by the camera, it first triggers the lidar. The lidar, getting the first 3D pose measurement, initializes a new track in the EKF. At this point the current state becomes LIDAR TRACKING and the tracking with lidar starts. The lidar reports detections for a few seconds (3 s in current tests) to ensure that not only the pose but also the speed of the drone is measured, and then stops tracking moving to the default position. Then the state is changed to CAMERA TRACKING, and tracking of the target with the camera continues, updating the EKF with the angle measurements from the camera. The errors between the angles predicted by the EKF in camera coordinates and the new angles of the target in the camera are continuously monitored, and when it exceeds 10 degrees, it is assumed that the drone has changed its track considerably. Then the PTU of the lidar is commanded to the new angles, detecting the drone again and updating the Kalman filter accordingly. This process is also done when the entries of the covariance matrix of the state, Σ, exceed a preset threshold.

The function to predict the states $\vec{x} = (x, y, z, \dot{x}, \dot{y}, \dot{z})$ follows the target motion according to the discrete time model given as

$$\hat{\vec{x}}_{t|t-1} = F_t \hat{\vec{x}}_{t-1|t-1} \tag{6}$$

[1] https://pointclouds.org/

where the transition matrix of the state F in discrete time is

$$F = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

Δt is the time between two consecutive pointclouds. Covariance estimate Σ_t at time t to predict the target position is given by

$$\Sigma_{t|t-1} = F_t \Sigma_{t-1|t-1} F_t^{\mathrm{T}} + Q_{t-1} \tag{8}$$

where Q models process covariance and accounts for uncertainty not taken into account by Σ_t. When a new centroid

$$z = \begin{bmatrix} x_{\text{measurement}} \\ y_{\text{measurement}} \\ z_{\text{measurement}} \end{bmatrix} \tag{9}$$

is detected by the lidar the update estimate is activated and the first step is corrected introducing an innovation term and a Kalman gain continuously evaluated by the filter at each step:

$$\begin{aligned} e_t &= z_t - h(\hat{\vec{x}}_{t|t-1}) && (10) \\ S_t &= H_t \Sigma_{t|t-1} H_t^{\mathrm{T}} + R_t && (11) \\ K_t &= \Sigma_{t|t-1} H_t^{\mathrm{T}} S_t^{-1} && (12) \end{aligned}$$

The matrix S estimates the uncertainty in the measurements, the actual covariance Σ_t and the actual state are updated with the innovation term and the Kalman gain

$$\begin{aligned} \hat{\vec{x}}_{t|t} &= \hat{\vec{x}}_{t|t-1} + K_t e_t && (13) \\ \Sigma_{t|t} &= (I - K_t H_t) \Sigma_{t|t-1} && (14) \end{aligned}$$

$h(\hat{\vec{x}}_t)$ is linearized around the current state estimate as H, a 3x6 matrix with entries 1 on the main diagonal for the lidar update because the measurements z are the three Cartesian coordinates. For the camera update H is a 2x6 matrix, where the measurements are the angles between x and y and between x and z. So the entries of H are given by the partial derivatives of the angles with respect to the states (Jacobian matrix). The measurement vector that is used for the camera update is

$$z = \begin{bmatrix} \theta_{\text{pan}} - {}_{\mathrm{C}}\Theta_{\mathrm{D}} \\ -\theta_{\text{tilt}} + {}_{\mathrm{C}}\Phi_{\mathrm{D}} \end{bmatrix} \tag{15}$$

The signs of this equation are set knowing that the pan angle between pan frame and world frame follows a counterclockwise convention, instead the azimuth and elevation calculated in the camera frame are negative moving to the left, positive to the right. Note that azimuth and elevation are defined in accordance with equations 4 and 5 and Fig. 2.

D. Lidar Tracking Angle Calculation

Since the camera cannot measure depth, only the direction along which the object was seen is monitored as the corresponding azimuth and elevation angles. However, the lidar sensor and the camera are not concentric. Hence, the lidar cannot directly align with the azimuth and elevation angles of the drone to measure the distance. This presents a challenge in determining the correct target angle to move the PTU of the lidar. To overcome this issue, two points were selected in the camera frame along the direction of the target as reported in the camera frame. Then, the angles of these two points with respect to the PTU of the lidar are calculated analytically, using the known geometry of the PTUs (Fig. 3). The two points are selected as the minimum range (R_{min}) and maximum range (R_{max}) the drone is expected to be seen.

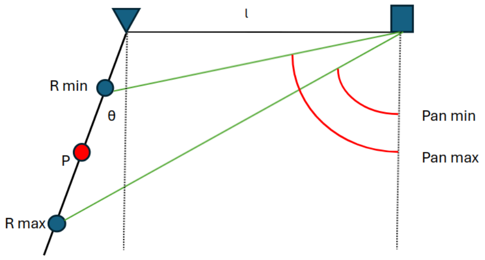


Fig. 3: Diagram of the detection system. The triangle represents the camera, and the rectangle represents the lidar. The drone is observed along the black line as the red circle P.

In the configuration shown in Fig. 3, the two points R_{min} and R_{max} lying along the direction of the drone, shown by the point P, allow us to calculate the range of pan angles necessary to move the lidar-PTU to detect the drone. These angles are given by

$$\bar{\theta}_{\text{pan-min}} = -\arctan\left(\frac{l - R_{max}\sin(\theta)}{R_{max}\cos(\theta)}\right), \tag{16}$$

$$\bar{\theta}_{\text{pan-max}} = -\arctan\left(\frac{l - R_{min}\sin(\theta)}{R_{min}\cos(\theta)}\right), \tag{17}$$

where l is the distance between the pan frame of the PTU of the lidar and the world frame (camera PTU). The arctan is negative to comply with the command input required to move the PTU correctly.

IV. Experimental Setup

The experimental setup consists of a tracking system and a drone. The tracking system contained a Velodyne VLP16 and a 5MP camera that were mounted on individual pan-tilt units, which were fixed at a distance of 0.70 meters on an aluminum bar as shown in Fig. 4. The Velodyne has 16 vertical beams, separated by 2°, with a horizontal resolution varying from 0.1° to 0.4°, and a corresponding frame rate varying between 5 and 20 Hz. In this work 10 Hz was used. The camera had 5MP resolution and a 5 mm lens, giving a FoV reaching 75°. The camera was initially calibrated, and then all the images received from the camera were rectified. In this work a Sky Hero Spyder X4 drone, which has a carbon fiber frame of size 0.85 m, was used as the target drone to detect (Fig. 5a). The code for the algorithms was programmed as C++ nodes for Robot Operating System (ROS).

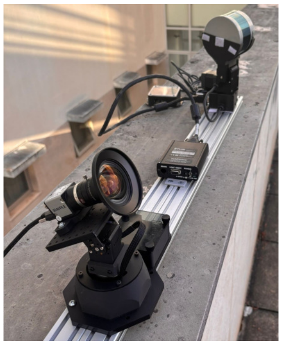
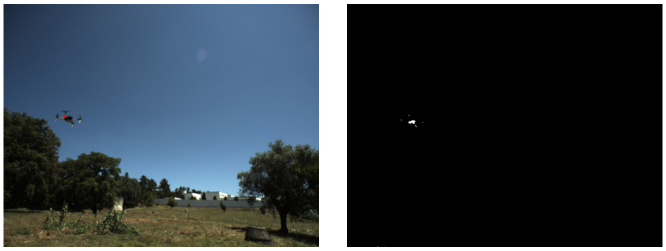


Fig. 4: Drone detection setup with the Velodyne VLP16 on a PTU 46-17.5 and the camera on a PhantomX XL430



(a)

(b)

Fig. 5: (a) Detected drone, marked with a red dot (b) The corresponding binary mask

V. Results and Discussions

The proposed approach was validated with multiple tests. Thanks to the large FoV of the camera, it was possible to cover a large area keeping the camera fixed. This allowed for saving a single background image in the beginning of each test, which lasted several minutes. Background subtraction followed by thresholding allowed the detection of the drone in many of the tests (Fig. 5). However, the method also started to report false detections when moving clouds were present in the background. Additionally, detection became unreliable in low light conditions when the drone had the trees as the background.

Fig. 6 shows the path of the drone (black) and the corresponding detections (blue) and EKF estimates (magenta) in one of the tests. The EKF updated coordinates follow correctly the path of the drone. At the beginning the drone goes up, the camera begins detecting when it reaches a height of 2 meters. Immediately after, the lidar starts to look in that direction and starts an EKF to track the state of the drone.

The transitions of the states for the camera based detector and lidar based detector as well as the estimates of the EKF are shown in Fig. 7 for a subset of a test. The azimuth angles of the drone with respect to the camera in the same time interval are shown in Fig. 8. The *Init* (INITIALIZATION) phase, which is the time between seeing in the camera and starting detecting with the lidar can be seen to be brief, but not always instantaneous. This is a side effect of the sparsity of the beams as well as the time it takes for the lidar to align with the angle. Lidar tracking is 3 s in this test, and the EKF can be seen to closely match the clusters (green zones). During camera tracking (yellow zone), the tracking performance can be seen to vary both in time and across the different axes. For example between 37 s and 41 s, camera based EKF follows the drone in the x-axis and y-axis closely, with errors less than 1m. However, in the same time interval, the error along the z-axis quickly reaches 1.3m. A consistent error can be observed along the z-axis when EKF relies only on camera. In Fig. 8, it can be seen that there is almost always an error between the azimuth angle of the drone predicted by the EKF during camera only tracking, and the azimuth angle measurement of the camera. This error is mainly caused by the motion of the drone.

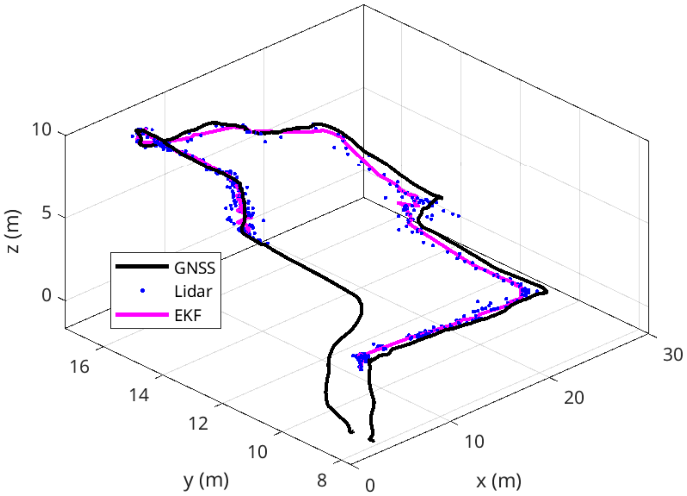
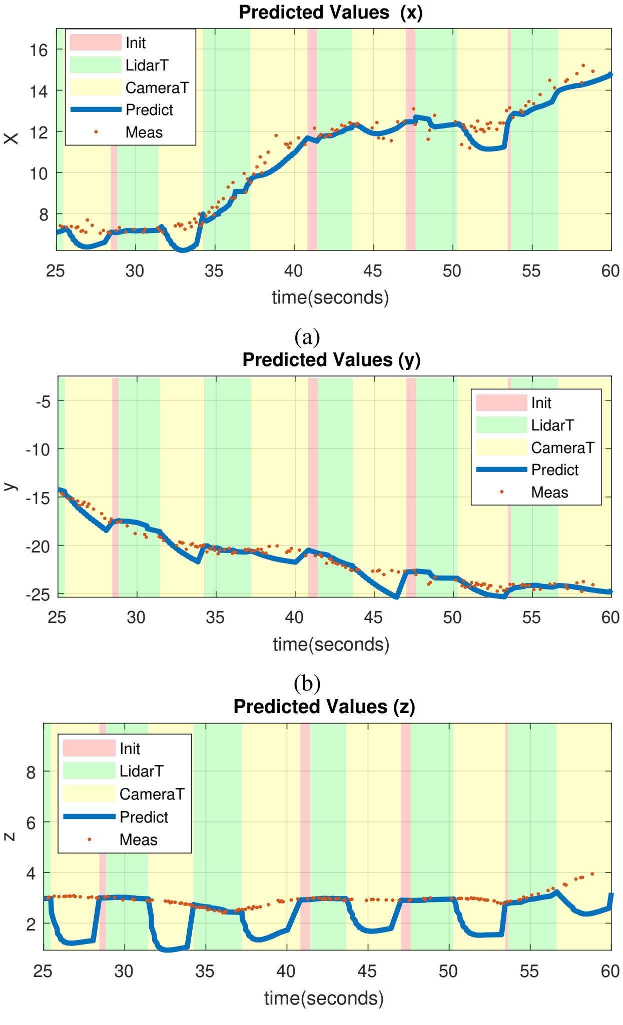


Fig. 6: 3D view of the path of the drone

VI. Conclusion

In this paper we presented a system for drone (UAV) detection and tracking using data from lidar and camera sensors, fusing both via an Extended Kalman filter. The fusion of the two sensors has enabled a more robust detection system for drones. Cameras are faster to detect changes in the sky compared to a 3D lidar, which has sparse beams. However, the camera lacks the capability to accurately localize a drone in the sky. In this work we fused the advantages of these two sensors. The experimental results have demonstrated that background subtraction for the camera works acceptably well in static conditions. Additionally, we showed a method to find the sweeping range for the lidar through the angles found by the camera and the geometric configuration. The method was effective in localizing targets with the lidar, helping achieve good localization. As future work we are planning to use a more robust drone detection algorithm based on YOLO instead of the current background subtraction method. This will allow validation of the approach in dynamic environments.



(a)

(b)

(c)

Fig. 7: Output of the EKF for x,y,z and the state of the system at the time.

References

[1] J. P. Škrinjar, P. Škorput, and M. Furdić, "Application of unmanned aerial vehicles in logistic processes," in *New Technologies, Development and Application*, I. Karabegović, Ed. Cham: Springer International Publishing, 2019, pp. 359–366.

[2] V. Chamola, P. Kotesh, A. Agarwal, Naren, N. Gupta, and M. Guizani, "A comprehensive review of unmanned aerial vehicle attacks and neutralization techniques," *Ad Hoc Networks*, vol. 111, p. 102324, 2021.

[3] A. Rozantsev, V. Lepetit, and P. Fua, "Flying objects detection from a single moving camera," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 4128–4136.

[4] F. Christnacher, S. Hengy, M. Laurenzis, A. Matwyschuk, P. Naz, S. Schertzer, and G. Schmitt, "Optical and acoustical UAV detection," in *Electro-Optical Remote Sensing X*, vol. 9988, 2016, p. 99880B.

[5] G. C. Birch and B. L. Woo, "Counter unmanned aerial systems testing: Evaluation of VIS SWIR MWIR and LWIR passive imagers." Sandia National Lab. (SNL-NM), Albuquerque, NM (United States), Tech. Rep. SAND2017-0921, Jan. 2017.

[6] T. Müller, "Robust drone detection for day/night counter-UAV with static VIS and SWIR cameras," in *Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR VIII*, vol. 10190, 2017, p. 1019018.

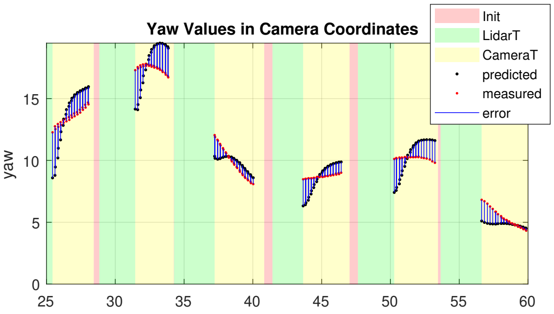


Fig. 8: Angle of the target as predicted by the EKF and as seen by the camera, with respect to the camera frame.

[7] S. Dogru and L. Marques, "Drone detection using sparse lidar measurements," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3062–3069, 2022.

[8] I. Catalano, H. Sier, X. Yu, T. Westerlund, and J. P. Queralta, "UAV tracking with solid-state lidars: Dynamic multi-frequency scan integration," in *2023 21st International Conference on Advanced Robotics (ICAR)*, 2023, pp. 417–424.

[9] H. Sier, X. Yu, I. Catalano, J. P. Queralta, Z. Zou, and T. Westerlund, "UAV tracking with lidar as a camera sensor in gnss-denied environments," in *2023 International Conference on Localization and GNSS (ICL-GNSS)*, 2023, pp. 1–7.

[10] Á. D. de Quevedo, F. I. Urzaiz, J. G. Menoyo, and A. A. López, "Drone detection with X-band ubiquitous radar," in *2018 19th International Radar Symposium (IRS)*, June 2018, pp. 1–10.

[11] S. Dogru and L. Marques, "Pursuing drones with drones using millimeter wave radar," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4156–4163, 2020.

[12] W.-C. Hu, C.-H. Chen, T.-Y. Chen, D.-Y. Huang, and Z.-C. Wu, "Moving object detection and tracking from video captured by moving camera," *Journal of Visual Communication and Image Representation*, vol. 30, pp. 164–180, 2015.

[13] M.-N. Chapel and T. Bouwmans, "Moving objects detection with a moving camera: A comprehensive review," *Computer Science Review*, vol. 38, p. 100310, 2020.

[14] C.-M. Huang, Y.-R. Chen, and L.-C. Fu, "Real-time object detection and tracking on a moving camera platform," in *2009 ICCAS-SICE*, 2009, pp. 717–722.

[15] J. Shi and Tomasi, "Good features to track," in *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 1994, pp. 593–600.

[16] Y. Jiang, L. Zhang, Z. Miao, X. Zhu, J. Gao, W. Hu, and Y.-G. Jiang, "Polarformer: Multi-camera 3D object detection with polar transformer," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 1, pp. 1042–1050, Jun. 2023.

[17] M. Hammer, M. Hebel, M. Laurenzis, and M. Arens, "Lidar-based detection and tracking of small UAVs," in *Emerging Imaging and Sensing Technologies for Security and Defence III; and Unmanned Sensors, Systems, and Countermeasures*, G. S. Buller, R. C. Hollins, R. A. Lamb, and M. Mueller, Eds., vol. 10799, International Society for Optics and Photonics. SPIE, 2018, p. 107990S.

[18] P. Wei, L. Cagle, T. Reza, J. Ball, and J. Gafford, "Lidar and camera detection fusion in a real-time industrial multi-sensor collision avoidance system," *Electronics*, vol. 7, no. 6, 2018.

[19] L. Caltagirone, M. Bellone, L. Svensson, and M. Wahde, "Lidar–camera fusion for road detection using fully convolutional neural networks," *Robotics and Autonomous Systems*, vol. 111, pp. 125–131, 2019.

[20] L. Liu, J. He, K. Ren, Z. Xiao, and Y. Hou, "A lidar–camera fusion 3D object detection algorithm," *Information*, vol. 13, no. 4, 2022.

[21] F. Zhang, D. Clarke, and A. Knoll, "Vehicle detection based on lidar and camera fusion," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2014, pp. 1620–1625.

[22] OpenCV Documentation Team. (2024) Camera calibration and 3D reconstruction. Accessed: July 21, 2024. [Online]. Available: https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html